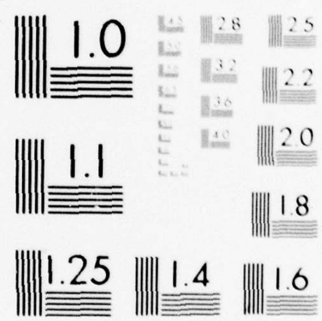
AD-A077 848 LOGICON INC LEXINGTON MA F/G 5/1
REQUIREMENTS STANDARDS STUDY. VOLUME II. TECHNICAL REPORT.(U)
OCT 79 D G SMITH, P B MERRITHEW F30602-77-C-0207
UNCLASSIFIED DSJ-R79007-VOL-2 RADC-TR-79-240-VOL-2 NL

1 OF 3
AD
A077848


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

LEVEL III (12)

A077847

AD A077848

RADC-TR-79-240, Vol II (of three)
Final Technical Report
October 1979

REQUIREMENTS STANDARDS STUDY
Technical Report

LOGICON

Daniel G. Smith
Paul B. Merrithew

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

REQUIREMENTS STANDARDS STUDY, VOL II

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 12 7 057

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-240, Vol II (of three) has been reviewed and is approved for publication.

APPROVED:

MICHAEL LANDES
Project Engineer

APPROVED:

WENDALL C. BAUMAN, COLONEL, USAF
Chief, Information Sciences Division

FOR THE COMMANDER:

JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 18 RADC-TR-79-240-VOL-2 II (of three)	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) 6 REQUIREMENTS STANDARDS STUDY. Volume II. Technical Report.	5. TYPE OF REPORT & PERIOD COVERED 9 Final [redacted] Report. Oct 77 — Jan 79,
	6. PERFORMING ORG. REPORT NUMBER 14 DSJ-R79007-VOL-2
7. AUTHOR(s) 10 Daniel G. Smith, Paul B. Merrithew	8. CONTRACT OR GRANT NUMBER(s) 15 F30602-77-C-0207
9. PERFORMING ORGANIZATION NAME AND ADDRESS Logicon, Inc. 18 Hartwell Avenue Lexington MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 16 55811807 17 18
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (ISIE) Griffiss AFB NY 13441	12. REPORT DATE 11 Oct[redacted]979
	13. NUMBER OF PAGES 196
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same 12 196	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Michael Landes (ISIE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Requirements Engineering
Requirements Definition and Analysis
System Specification
Functional Specification
Software Specification
Development Specification
Requirements Specification Language
Requirements Engineering Standards

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is one of three volumes which provide information and guidance for defining and analyzing military system/segment specification and development specification requirements (MIL-STD-490/483 (USAF)). Volume I is a technical overview describing the purpose of the study, the technical approach and a history of Air Force systems engineering management. It summarizes the results of the study: a Requirements Engineering Guidebook, an example using the Guidebook, associated automated tool capabilities, automated tool design (Cont'd)

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409 990 Jm

Item 20 (Cont'd)

approaches, and recommendations on implementing the Guidebook in the Air Force systems acquisition life cycle. Volume II expands upon the material summarized in the first volume. Volume III is the Requirements Engineering Guidebook. The Requirements Engineering Guidebook describes the characteristics of good requirements, the various system requirements engineering procedures are described in the form of a procedural flow with accompanying guidelines and standards for performing fourteen requirements engineering activities. Each requirements engineering activity is supplemented by a description of the specific issues to be addressed during the first two phases of the Air Force acquisition life cycle - the Conceptual and Validation Phases.

PREFACE

This report is one of three volumes prepared to assist government and contractor personnel in managing and performing system requirements definition and analysis: requirements engineering. The primary results of this study has been the definition of guidelines and standards for requirements engineering (Requirements Engineering Guidebook) and the identification of automated aids to support the application of the guidelines and standards during the initial phases of the Air Force system acquisition life cycle - the Conceptual and Validation Phases.

This study reflects Logicon's experience with an automated requirements engineering tool applied in support of the acquistion of a large Air Force surveillance system. The Requirements Engineering Guidebook reflects the needs of an Air Force System Program Office acquisition environment; however, the basic requirements engineering principles and guidance are easily adapted to other acquisition environments.

This report was prepared by Logicon for the Air Force Systems Command (AFSC), Rome Air Development Center (RADC), Software Engineering Section. Administrative review and technical coordination of this report have been accomplished for RADC by Mr. Michael Landes (project officer).

Review of this report was accomplished at RADC, by Electronic Systems Division (AFSC/ESD) personnel at Hanscom, AFB, and by Logicon personnel. Special thanks to the many reviewers and for the patience and skills of Ms. Marcia Brehm and Ms. Deborah Queen for the technical typing, proofing, and revisions.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

SECTION 1 INTRODUCTION

1.1 Study Objectives

The objectives of the Requirements Standard Study were (1) to define a set of standards for the definition and analysis of the requirement of a system during the initial phases of the acquisition life cycle, and (2) to identify the functional requirements for automated tools intended to support the standard by providing assistance in the definition and analysis of the system requirements and by assuring compliance with the standard.

1.2 Project Background

The development of reliable and easy-to-maintain software demands concern for quality at each phase of system acquisition. Because of the high visibility of problems discovered during the later phases of development, coding, testing and integration of the system have received considerable scrutiny. An extensive amount of research and technical direction has been focused on these later phases in an effort to enhance the quality of the system and reduce cost and schedule overruns. Most of the attention has been focused upon coding methodology, with some emphasis on design standards such as top-down design and implementation. However, studies indicate that errors caused by ill-defined and poorly documented requirements are the most expensive to fix. As the comlexity of systems has increased, the volume of system requirements and associated requirements documents (specifications, etc.) have exceeded the comprehension of the engineering disciplines and techniques employed.

In recent years, a number of private and government projects have addressed the need for quality in requirements definition and analysis during the initial phases of the system acquisition[1]. The application of automated techniques to assist in requirements engineering activities and to document

[1] Throughout this report, requirements definition and analysis activities are collectively referred to as requirements engineering.

requirements has also advanced. The state of the art has advanced to the point where the need for improved guidance for requirements engineering is practical with the aid of automated tools. This study provides guidelines and standards for requirements engineering in the form of a guidebook (Volume III) and identifies the automated tool capabilities needed to implement the guidebook. The study results provide additional guidance to increase the quality and to minimize the life-cycle costs of systems developed for the Air Force. The application of standards in coding has already resulted in measurable improvements in the quality of software. The requirements engineering guidebook produced during this study is designed to supplement these improvements by providing similar guidance for the early phases of system requirements definition and analysis.

1.3 Contents

The remainder of this volume consists of four sections and five appendices as follows:

- Section 2 - Technical Approach. Provides a technical description of the seven tasks performed during this study.

- Section 3 - Air Force System Engineering Management Describes the trend of Air Force systems engineering management over the past decades and identifies present practices.

- Section 4 - Summary of Study Results. Provides a summary of the Requirements Engineering Guidebook presented in Volume III and describes an approach for implementing the guidebook. In addition, this section contains an example of the application of the guidebook, the list of functional capabilities required for an automated tool to support the guidebook, and concludes with a description of two approaches for applying an existing automated requirements engineering tool.

- Section 5 - Results and Recommendations. Provides a summary of conclusions based upon the discussions in the previous sections. In addition, this section provides a list of recommendations concerning the implementation of the guidebook, use of automated tools and additional areas for future research.

- Appendix A - References. Provides a list of selected references which are pertinent to the discussions contained in the previous sections.

- Appendix B - Glossary. Provides definitions of the major terms used in Air Force system acquisition and concludes with a list of acronyms and abbreviations.

- Appendix C - Bibliography. Lists selected books, papers, and military regulations, specifications, and standards that are pertinent to the study objectives.

- Appendix D - Automated Requirements Engineering Tool Capabilities List. Provides a concise list of capabilities needed in automated requirements engineering tools.

- Appendix E - Requirements Engineering Example. Provides an example of the application of the Requirements Engineering Guidebook in conjunction with an automated requirements engineering tool. This appendix has been prepared for inclusion in the RADC Computer Software Development Specification.

SECTION 2 TECHNICAL APPROACH

2.1 Overview of Technical Tasks

This study was organized into seven tasks as illustrated in Figure 1.

Tasks 1, 2 and 3 defined a set of requirements engineering guidelines and standards for the initial phases of the Air Force system acquisition life cycle - the Conceptual Phase and the Validation Phase. The guidelines and standards are presented in Volume III, titled Requirements Engineering Guidebook. This separate presentation was selected to provide a complete and concise document which is more readily usable for requirements engineering purposes by Air Force program offices and for potential contracting purposes.

Tasks 5 and 6 defined the functional requirements of an automated requirements engineering tool which would support the application of the Requirements Engineering Guidebook and identified two approaches for implementing the tool. The list of tool capabilities is described in 4.2, the approaches are described in 4.5.

Tasks 4 and 7 developed a description of the impacts of the Requirements Engineering Guidebook on existing Air Force practices, regulations, standards, and specifications and prepared the final report for the study. The description of impacts is presented in 4.4. The final report contains a review of the history of requirements engineering in Air Force, presents the Requirements Engineering Guidebook, descibes the guidebook's implementation impacts and requirements, and provides numerous results and recommendations on the guidebook, automated tool approaches, and areas of further research and development.

2.2 Tasks Inputs

The seven tasks described above and other tasks were accomplished

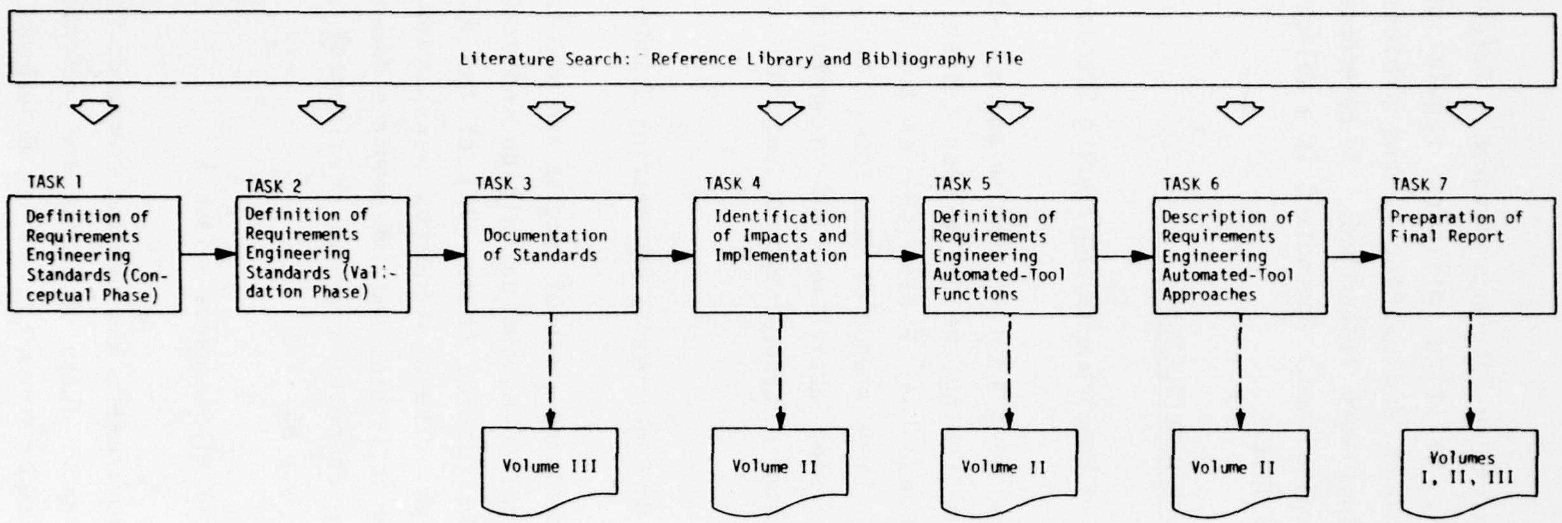


Figure 1. Requirements Standards Study Technical Approach

concurrently with an extensive literature search. Defense Documentation Center project files, existing military regulations, standards, specifications, guidebooks and handbooks, and professional journals and conference proceedings were researched. A reference library and a bibliography file were maintained. Appendix C is a selective bibliography generated from the study file.

2.3 Detailed Task Description

2.3.1 Definition of Requirements Engineering Standards (CP) - Task 1

Task 1 concentrated on defining a set of concise and well-defined standards for managing and accomplishing the definition and analysis of system requirements in the Conceptual Phase (CP) of the Air Force system acquisition life cycle. These standards were specifically oriented toward the requirements engineering activities and intermediate documentation leading to the preparation of system/segment specifications, MIL-STD-490/483 (USAF), Type A.

2.3.2 Definition of Requirements Engineering Standards (VP) - Task 2

Task 2 concentrated on defining a set of concise and well-defined standards for managing and accomplishing the definition and analysis of system requirements in the Validation Phase (VP) of the Air Force system acquisition life cycle. These standards were oriented toward the requirements engineering activities and intermediate documentation leading to the preparation of computer program development specifications, MIL-STD-490/483 (USAF), Type B5.

2.3.3 Documentation of Standards - Task 3

Task 3 documented the requirements engineering standards for the Conceptual Phase and Validation Phase. These standards are presented in Volume III of this report - the Requirements Engineering Guidebook. The standards

describe the charateristics of quality requirements[1], the types of system requirements, and the requirements engineering procedures. The requirements engineering procedures are described in the form of a general procedural flow with accompanying text which describes the requirements engineering activities to be performed. Each requirements engineering activity is supplemented by a description which addresses the specific issues to be addressed during the first two phases of the Air Force system acquisition life cycle - the Conceptual and Validation Phases.

2.3.4 Identification of Impacts and Implementation Approach - Task 4

In Task 4 existing Department of Defense and Air Force regulations, standards, and specifications were identified and evaluated to determine any points of conflict between current practices and the Requirements Engineering Guidebook. A description of the approach for implementing the guidebook in an Air Force program office was also prepared. The impacts and implementation description are presented in 4.4.

2.3.5 Definition of Requirements Engineering Automated-Tool Functions - Task 5

In Task 5 a list of essential tool capabilities which support the definition and analysis activities of the guidebook were defined. This list, as presented in 4.2, includes the language features and analyzer requirements of the tools. The analyzer requirements were grouped into six general categories: requirements data base management, functional analysis, input /output analysis, traceability analysis, test analysis, and documentation.

[1] The term 'quality requirements' is used throughout this study to denote system requirements which are complete, consistent, testable, and traceable. This characteristic is the result of the requirements being discretely identified and well-organized as discussed in this report.

2.3.6 Description of Requirements Engineering Automated-Tool Approaches - Task 6

During Task 6 existing requirements engineering automated tools were reviewed. Two approaches were identified for applying an existing Air Force requirements engineering tool to satisfy the requirements engineering tool functions identified in Task 5. The tool review provided the basis upon which the two approaches were evaluated for feasibility, performance, and cost-effectiveness.

2.3.7 Preparation of Final Report - Task 7

The final report, prepared during Task 7, presents a summary of the entire study. This report document (Volume II) contains the complete description of the technical results, including the results of the previous tasks: 4, 5 and 6. Appendix E provides an example of the application of the Requirements Engineering Guidebook to the requirements definition and analysis activities of an Air Force surveillance system acquisition; this example was prepared for inclusion in the RADC Computer Program Software Development Specification. Volume III is the Requirements Engineering Guidebook and Volume I is a technical overview of this volume.

SECTION 3 AIR FORCE SYSTEMS ENGINEERING MANAGEMENT

3.1 Introduction

The complexity of military systems development has continued to outpace the management and technical resources supporting the acquisition process. The decline in DoD manpower resources over recent years has had a two-fold effect on military systems development. The higher reliance on automated systems has been constantly increasing over the past decades and has been heightened in recent years by the desire to reduce the system's operational crew requirements and associated support functions. In conjunction with decreasing operational personnel, the system implementing agencies of the military have also been impacted by staff reductions. The net effect is that systems are rising in complexity at the same time that military engineering resources responsible for the acquisition have been effectively reduced to managers of the engineering efforts. The lack of engineering staff resources is even further complicated by the lack of well-defined system engineering approaches for managing and accomplishing quality systems requirements definition and analysis.

3.2 Initial Regulated Approach

During the 1960s and into the early 1970s, systems development in the Air Force Systems Command (AFSC) was regulated by a series of manuals, the AFSCM 375 series. This series was adapted primarily from industry system engineering procedures and integrated into formal guidance for Air Force government program offices and associated contractors. Two of these manuals concentrated on system engineering procedures and documentation requirements. The systems engineering volume (AFSCM 375-5 [1]) contained a procedural flow diagram and associated text which delineated the system engineering activities, intermediate documentation requirements, formal project reviews and other requirements which were to be accomplished throughout the system development life cycle. Keyed to the 375-5 system engineering procedural volume was the first volume in the series which addressed specification requirements, AFSCM 375-1 [2]. This highly regulated approach was necessitated by the increasing delivery of obsolete

systems, which resulted from the less-regulated systems development approach of previous decades (1940s-1950s). Added to this obsolescence problem was the rising complexity of the systems being developed to meet the national defense needs of the post World War II decades.

The AFSCM 375 series provided for flexibility in its application but was not completely understood. As a result of the difficulties encountered in applying AFSCM 375, the Air Force began to rescind parts of the series during the late 1960s. A primary contributor to the decline of the series has been attributed to the tons of documentation which was developed and delivered to the Air Force for the C5-A aircraft [3]. The Air Force documentation requirements of AFSCM 375-1 evolved almost unchanged into the present tri-service military standard for specification practices, MIL-STD-490 [4]. MIL-STD-490 along with the additional requirements of the Air Force standard on configuration management practices, MIL-STD-483 (USAF) [5], represents an almost complete translation of AFSCM 375-1. The system engineering procedures of AFSCM 375-5 evolved into the present regulation for Air Force program office engineering (AFR 800-3 [6]). Contractor system engineering requirements were described in an initial version of the present Air Force standard for engineering management, MIL-STD-499A (USAF) [7].

3.3 Present Less-regulated Practices

As previously reported [8], it is becoming evident that the present military management practices (regulations, standards, and specification practices) and the technical resources are not adequate for the increasingly complex military systems currently being developed. In recent years the Department of Defense (DoD) and the Air Force introduced new directives, regulations, and practices targeted at specific system development problem areas. One of the most troublesome areas is software within military systems, most often called embedded software. As a result new regulations [9,10,11,12], guidebooks, and special research projects on modern programming practices for software implementation have been developed. The application of these new regulations and practices is

beginning to show beneficial results. However, other areas have received less attention. A principle area of deficiency has resulted from the inadequacies of the system engineering guidance in the present Air Force contractor and government standards and regulations, namely the previously mentioned MIL-STD-499A and AFR 800-3. In the wake of rescinding the system engineering requirements of AFSCM 375-5, significant practices for systems requirements engineering were not translated or updated into present practices. As a result many essential requirements engineering practices have been de-emphasized. Under current Air Force practices, the contractor may be required to propose a specific system engineering management plan (SEMP) tailored to the specific project and in compliance with MIL-STD-499A. The SEMP is then subject to the Air Force program office evaluation process. On the other hand, Air Force program office engineering tasks are even more sketchily defined in AFR 800-3. Even a brief review of these two documents reveals that the requirements engineering activities are not commensurate with the complexity of the management and engineering requirements of the systems being developed.

This trend toward less regulated Air Force systems engineering management has been encouraged by defense contractors in a desire to allow for more competitive and innovative approaches to systems development. Numerous contractors have responded by developing systems engineering procedures. However, other defense contractors and military agencies have not developed systems engineering or management practices which satisfy the real technical and management needs of Air Force programs.

3.4 Research and Directions

In recent years systems requirements engineering has received renewed attention within academic and military research and development (R&D) environments and is now coming to the forefront of research and applications for improved military systems development. Many defense contractors have participated directly in military R&D activities for developing and applying automated tools to the requirements engineering process. In recent years AFSC's Electronic Systems Division (ESD) has acquired a computer-aided requirements engineering tool, CADSAT, and has

encouraged the application of this computer aid in the requirements engineering activities in the early phases of Air Force systems development life cycle[1]. Logicon systems analysts have employed CADSAT for several years in defining and analyzing the system requirements for a large Air Force surveillance system. As a result of this application, Logicon has made extensions to CADSAT to satisfy requirements engineering management and technical needs. These extensions have made CADSAT more suitable to specific military systems acquisition activities. Concurrent with applying and extending CADSAT, Logicon has developed an approach for CADSAT's use. As with other research directed in recent years toward defining standards for modern programming practices in Air Force systems development, requirements engineering has been identified as a target for improved standardization. Within this renewed interest and based upon the surveillance system experience using CADSAT, Logicon began developing a requirements engineering guidelines and standards in 1977 under contract to the Rome Air Development Center (RADC), Griffiss AFB, New York. This study, titled the Requirements Standard Study (RSS), is the subject of the remainder of this report. The guidelines and standards are based upon Logicon's requirements engineering experience and the use of CADSAT. The primary results of the RSS are a Requirements Engineering Guidebook (Volume III) and the identification of automated aids to support the application of the guidebook during the Conceptual and Validation Phases. The guidebook has been developed in response to the requirements engineering inadequacies of the current Air Force system engineering procedures. However, the principles of the guidebook are applicable to other systems acquisition environments.

1 The Computer-Aided Design and Specification Analysis Tool (CADSAT) is an Air-Force-owned requirements analysis tool developed by the University of Michigan under ESD/TOI Contract F19628-76-C-0197. [13] [14]. The extended version is a modification developed by Logicon for applications to military systems under ESD/OCU Contract F19628-76-C-0218. CADSAT's User Requirements Language/User Requirements Analyzer is basically equivalent to the Problem Statement Language and Problem Statement Analyzer (PSL/PSA) developed at the University of Michigan. [15]

SECTION 4 SUMMARY OF STUDY RESULTS

4.1 Requirements Engineering Standard

4.1.1 *Requirements Engineering*

4.1.1.1 Lacking Definition: Current Air Force standards and regulations neither identify nor define requirements engineering separately from other engineering disciplines. During this study, requirements engineering was determined to be a distinct engineering discipline which needs to be addressed separately from other forms of engineering. During the 1960s requirements engineering was integral to the procedural aspects of the system engineering process established under AFSCM 375-5. For instance, the use of functional flow block diagrams and the documentation of associated requirements on special forms (requirement allocation sheets) were two AFSCM 375-5 requirements engineering activities. Neither the current military standard for engineering management (MIL-STD-499A) nor the guidance for program office engineering (AFR 800-3) define requirements engineering and, as described earlier, these two documents omit procedures entirely. AFR 800-3 identifies eleven engineering or technical functions under engineering management; these functions are performed by the program office throughout the system acquisition life cycle: systems engineering, design engineering, specialty engineering, test engineering, production engineering, logistics engineering, civil engineering, human factors engineering, configuration engineering, technical data control and technical program planning and control. The definitions of the first two functions, systems engineering and design engineering, indicate the degree to which requirements engineering is not a prominent engineering activity in current Air Force systems acquisition.

According to AFR 800-3, systems engineering covers a broad range of scientific and engineering efforts with three principle elements: functional analysis, synthesis, and trade studies or cost-effectiveness

optimization. System engineering tasks are only identified as follows: mission requirements analysis, functional analysis, requirements specification preparation. Since procedural aspects are omitted, the quality and performance of the requirements engineering activities are dependent upon individual or agency programs or initiatives. The second AFR 800-3 function, Design Engineering, is defined to be the activity of using the technical information (requirements, goals, criteria, constraints, etc.) developed through the system engineering process to develop detailed design approaches, design solutions, and the test procedures to prove the solutions. This second engineering management function clearly highlights that requirements engineering is integrated into the previously defined system engineering function. Requirements engineering is vaguely defined to be part of the system engineering process: the functional analysis-synthesis tasks. This type and form of guidance is inappropriate for the requirements engineering activities which must be accomplished during the early phases of the acquisition process.

4.1.1.2 Requirements Engineering Defined: A requirements engineering definition must be stated and the procedural issues addressed. The following definition has been prepared during the course of the RSS:

> Requirements engineering - An iterative process of defining the system requirements and analyzing the integrity of the requirements. This process involves all areas of system development preceding the actual design of the system. The products of the requirements engineering process can be evaluated for completeness, consistency, testability, and traceability. The essential goal of requirements engineering is to thoroughly evaluate the needs which the system must satisfy.

This definition distinguishes requirements engineering from other engineering management tasks such as program planning, costing, trade-off studies, and a host of other issues surrounding the early phase of systems development. The definition distinguishes requirements engineering as concentrating on the actual definition and analysis of the system requirements.

4.1.2 Requirements Engineering Goal

A system in the context of this presentation is an aggregate of equipment, personnel resources, capabilities and techniques which collectively perform an operational role. The composite system includes all related facilities, items, materials, services, and personnel required for the system's self-sufficient operational deployment. Requirements engineering concentrates upon defining the system as a whole in operational mission terms including associated performance requirements. In requirements engineering, the analyst must avoid orientation towards specific solutions by concentrating upon defining the system in terms of what must be accomplished. The lack of specific approaches and techniques for military requirements engineering allows even the best intentioned analyst to digress rapidly from the "need" category to the "how-to" or solution oriented requirements definitions. The results are "system requirements" which are semantically riddled with solution overtones or specific design details. The requirements engineering process must recognize this tendency and must allow for effective feedback into the process. The analyst must be aware that system documentation, such as functional (Type A, MIL-STD-490/483 (USAF) System/ Segment Specifications) and development (Type B, MIL-STD-490/483 (USAF) specifications, is the media for communicating the system requirements to the design engineers. The goal of requirements engineering is to identify discrete requirements of the system and to organize these requirements in effective ways for further analysis. The result of this process is a set of "quality requirements."

4.1.3 Quality Requirements Characteristics

4.1.3.1 Introduction: A set of quality requirements consists of discrete requirements, well organized to permit further analysis. Initial documentation for identifying user system requirements may include early planning documents and specifications for similar systems, for system interfaces, and for existing or previously defined subsystems. In addition, documentation derived from engineering studies and protyping or

experimental test systems may be available. If the engineering activities have advanced beyond the planning and study stage, specification documents such as Type A or Type B specifications may have already been developed. These early requirements documents usually have one prevailing characteristic: the requirements are spread over various source documents and/or presented in various parts of the the documents, and the requirements overlap each other. This is partly because of the fragmented nature of the early planning and study efforts which are formulative and investigatory. Another facto has been the lack of guidance in requirements engineering and the orientation of engineers to the specification documents. The specification documents in many instances are products written to meet acquisition needs and schedule rather than repositories of quality requirements.

4.1.3.2 Discrete Requirements: Figure 2 illustrates the first characteristic of quality requirements: discreteness. The key to identifying discrete requirements is to break the source documentation into individual parts which represent non-overlapping requirements. Requirements should then be categorized as functions the system must accomplish or as system constraints (performance, physical, operability, test, design). At this point missing or incomplete requirements can be more readily identified. This itemization and categorization of requirements introduces clarity, whereas the source documentation may be overstated, ambiguous, redundant, incomplete, and inconsistent. This itemization also provides the basis for verifying the quality of the requirements and the ability to test the requirements in the target system.

4.1.3.3 Organization of Requirements: The second characteristic of a good statement of requirements is the arrangement of the requirements in effective ways of additional analysis and for communicating these requirements to the using agency and to design engineers. The identification of discrete requirements provides some awareness of omissions and gaps in the requirements. This awareness is further heightened by organizing the requirements in various ways which identify all the relationships among the discrete requirements (Figure 2).

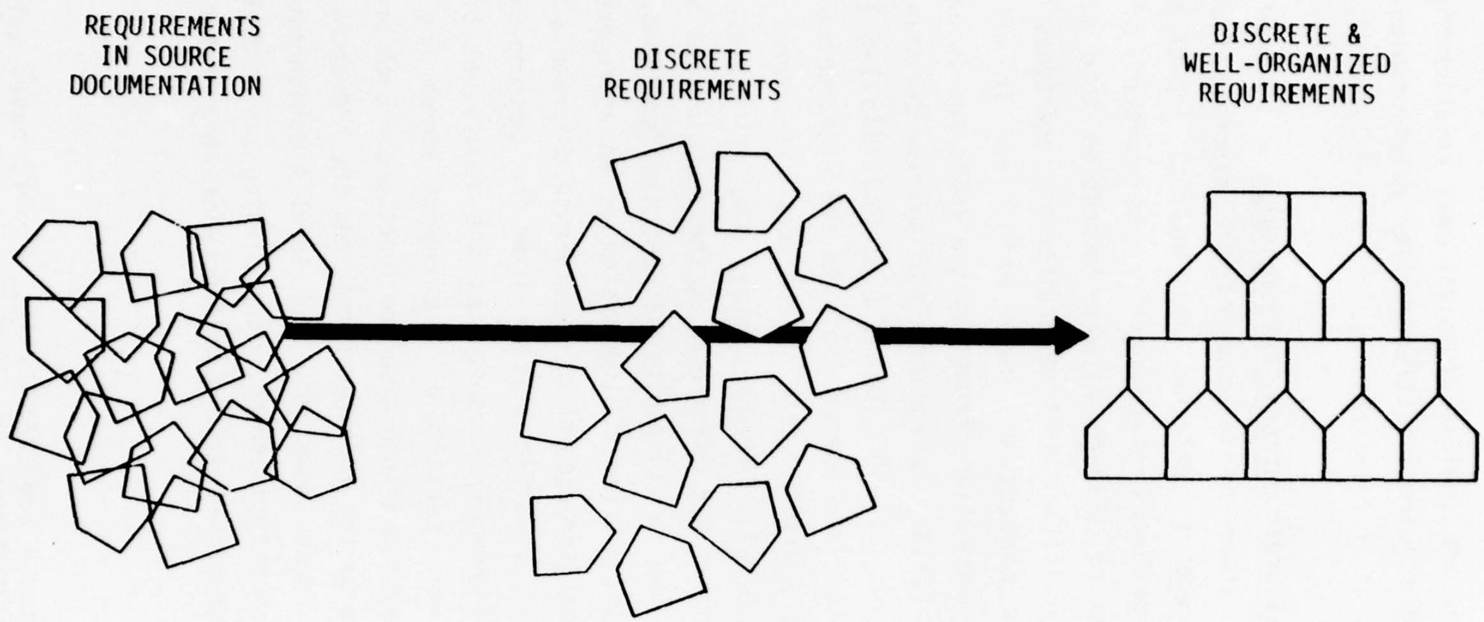
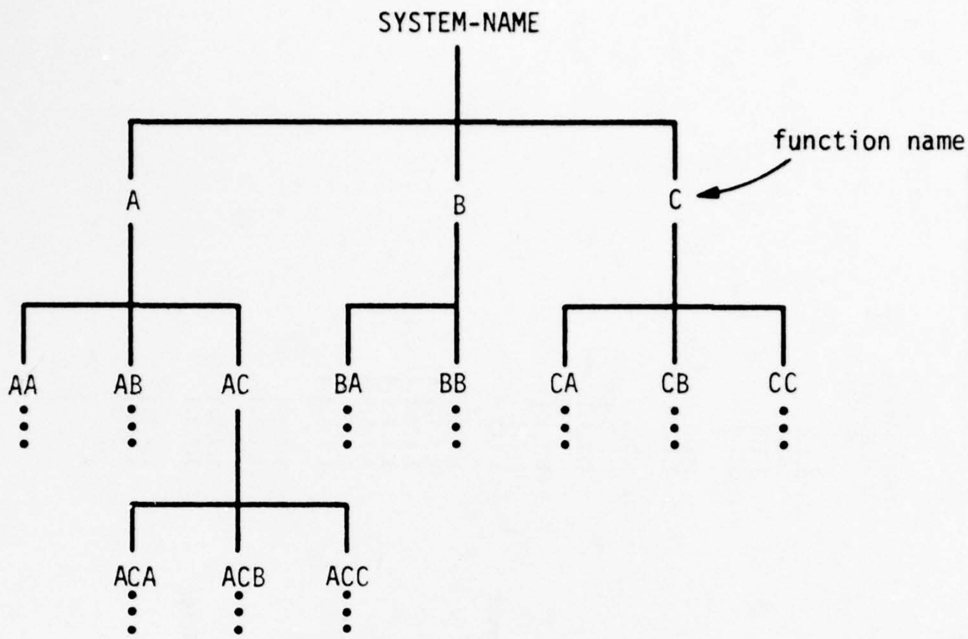


Figure 2. Development of Discrete and Well-Organized Requirements

These relationships are logical organizational relationships, system flow relationships, and requirements traceability relationships. The following paragraphs discuss these relationships.

4.1.3.3.1 Logical Organizational Relationships - Logical organizational relationships are shown by structuring the discrete functions and the information requirements (external and internal input/ output) of the system into hierarchical structures. The concept of a functional hierarchical structure (Figure 3) was introduced into military systems development through initial systems engineering practices dating back to the 1940s. This concept has been maintained in military systems development and documentation throughout the 1960s and is an integral part of the current military standards for system documentation, i.e., MIL-STD-490 [4], MIL-STD-483 (USAF) [5], DoD 7935.1-S [16]. Current techniques for system development, such as the Hierarchical Input-Output-Process (HIPO [17]), visual table of contents and automated requirements analysis tools (PSL/PSA, CADSAT), retain the principles of functional hierarchical structures for communicating the functions to be accomplished by the system and the relationships between the functions. This form of organization provides a view of the system as an aggregate of functions broken into a logical arrangement of subordinate discrete activities which must be performed. A sample portion from the Logicon-Extended CADSAT Structure Report (Figure 4) demonstrates the functional break out of a space system[1]. This section of the report shows the hierarchical breakdown of space-system-x into discrete functions. Each breakdown of the functions is denoted by the indented format and the hierarchy level number. For example, boost breaks down into four level 4 subfunctions. Over the course of requirements engineering, many missing or incomplete functions can be directly identified from the functional hierarchical structure.

[1] Figures 4, 8, and 9 are CADSAT-like reports based upon the space-system-x example contained in AFSCM 375-5, attachment 2, pp. 128-130 [1]



Graphic Representation

SYSTEM-NAME
A
AA ...
AB ...
AC ...
ACA ...
ACB ...
ACC ...
B
BA ...
BB ...
C
CA ...
CB ...
CC ...

Indented Representation

Figure 3. Functional Hierarchical Structure

SPACE SYSTEM X   LOGICON FUNCTIONAL HIERARCHICAL STRUCTURE REPORT   DATE 01/23/79   Page 1

LINE NO.	HIERARCHY LEVEL NO.	REQUIREMENT NAME	REF. DOCUMENT PARAGRAPH NO.
1	1	space-system-x	3.1.1
2	2	status-monitoring	3.1.1
3		triggers launch	3.1.1
4	2	launch	3.1.1
5		triggers flight-mission	3.1.1
6	2	flight-mission	3.7.3
7		triggers boost	3.7.3.1
8		triggers steer	3.7.3.1
9	3	boost	3.7.3.1
10		triggers release-payload	3.7.3.1
11		triggers stage-1-and-stage-2-thrust	3.7.3.1
12	4	stage-1-and-stage-2-thrust	3.7.3.1.4
13		triggers jettison-stage-1	3.7.3.1.4
14	4	jettison-stage-1	3.7.3.1.4
15		triggers stage-2-thrust	3.7.3.1.4
16	4	stage-2-thrust	3.7.3.1.4
17		triggers stage-2-shut-down	3.7.3.1.4
18	4	stage-2-shut-down	3.7.3.1.4
19	3	steer	3.7.3.1
20		triggers release-payload	3.7.3.1
21	3	release-payload	3.7.3.1
22		triggers decelerate-launch-vehicle	3.7.3.1
23	3	decelerate-launch-vehicle	3.7.3.1
24		triggers payload-coast	3.7.3.1

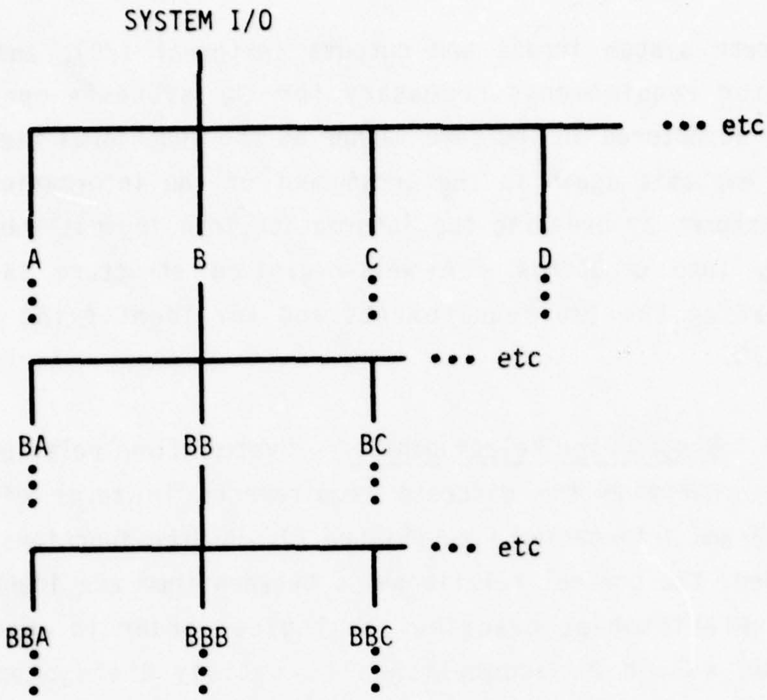
Figure 4. Space System X: Functional Hierarchical Structure

The discrete system inputs and outputs (external I/O), and the internal information requirements necessary for the system's operation can be logically structured in the same manner as the functional hierarchy (Figure 5). The emphasis again is the arrangment of the information requirements into structures by breaking the information into logical subordinate parts or simply into groupings. A well-organized structure is effective in communicating the I/O requirements and for identifying incomplete or missing I/O.

4.1.3.3.2 System Flow Relationships - System flow relationships can be shown by organizing the discrete requirements in terms of control flow (Figure 6) and information flow (Figure 7). As the functions of the system are defined, the control relationships between them are identified. These control relationships describe the logical order in which the system activities should be accomplished to satisfy the system mission and operational requirements. Figure 8 is a control-flow report for a portion of the space-system-x. In this report (CADSAT Process Chain) the flow of control is from left to right. Any number of CADSAT process chain reports can be generated to provide the analyst with a comprehensive understanding of the system control flow.

Control-flow analysis provides a means of viewing the system from an activity-oriented perspective and is often referred to as functional-flow analysis. As a result of this analysis, the requirements are viewed in a well-organized manner and missing or incomplete functions and relationships between the functions are identified. Final control-flow documentation becomes another effective means for communicating system requirements to design engineers.

On the other hand, information flow analysis builds upon the I/O hierarchical structure by providing a means of viewing the system from an information systems perspective. During this analysis the flow relationships between external system inputs and resulting outputs are identified (Figure 7). Quite often the most effective means of performing



Graphic Representation

SYSTEM I/O
INPUT-A ...
OUTPUT-B
BA ...
BB
BBA ...
BBB ...
BBC ...
(etc)
BC ...
(etc)
INPUT-C ...
OUTPUT-D ...
(etc)

Indented Representation

Figure 5. I/O Hierarchical Structure

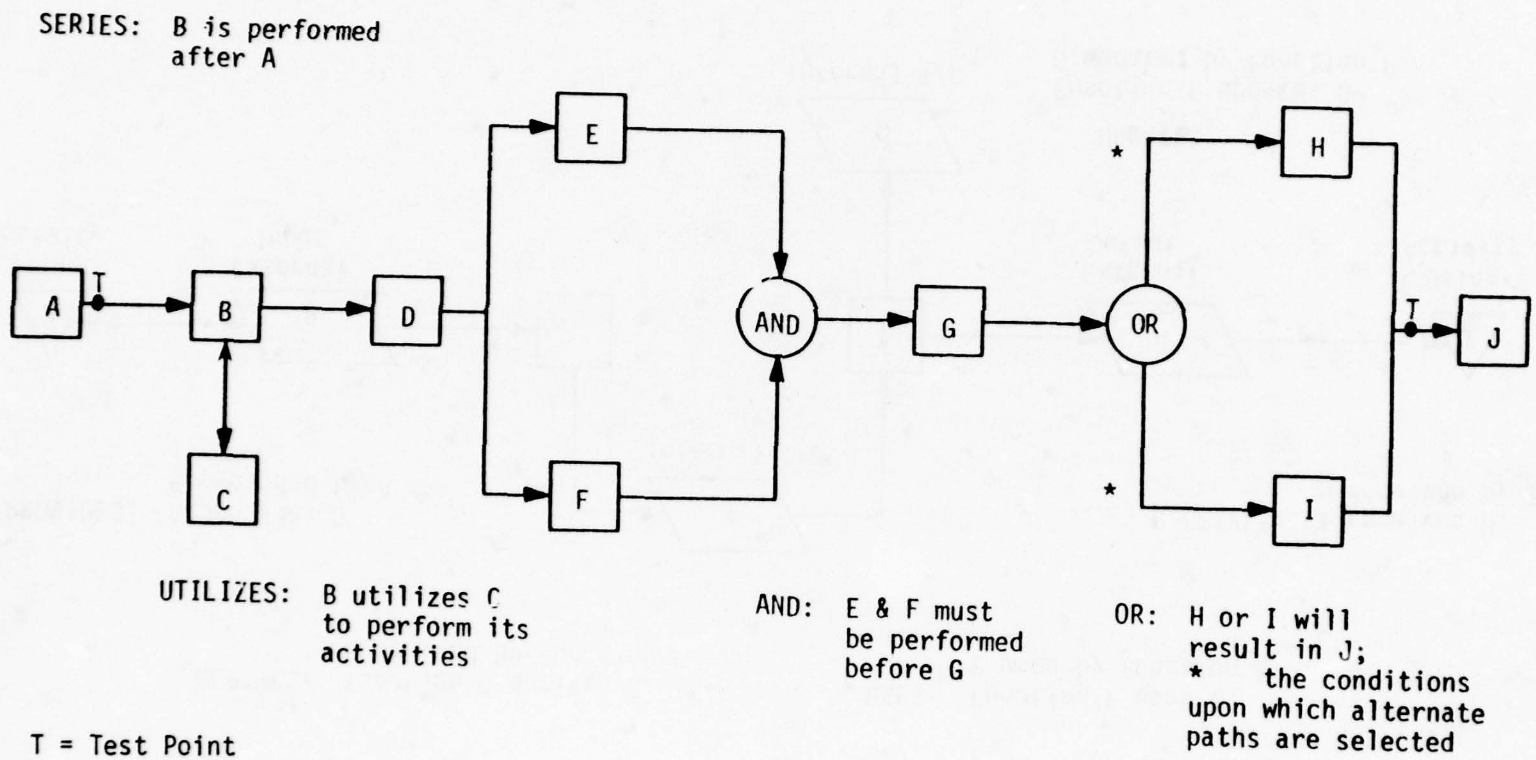


Figure 6. Control-Flow Diagram

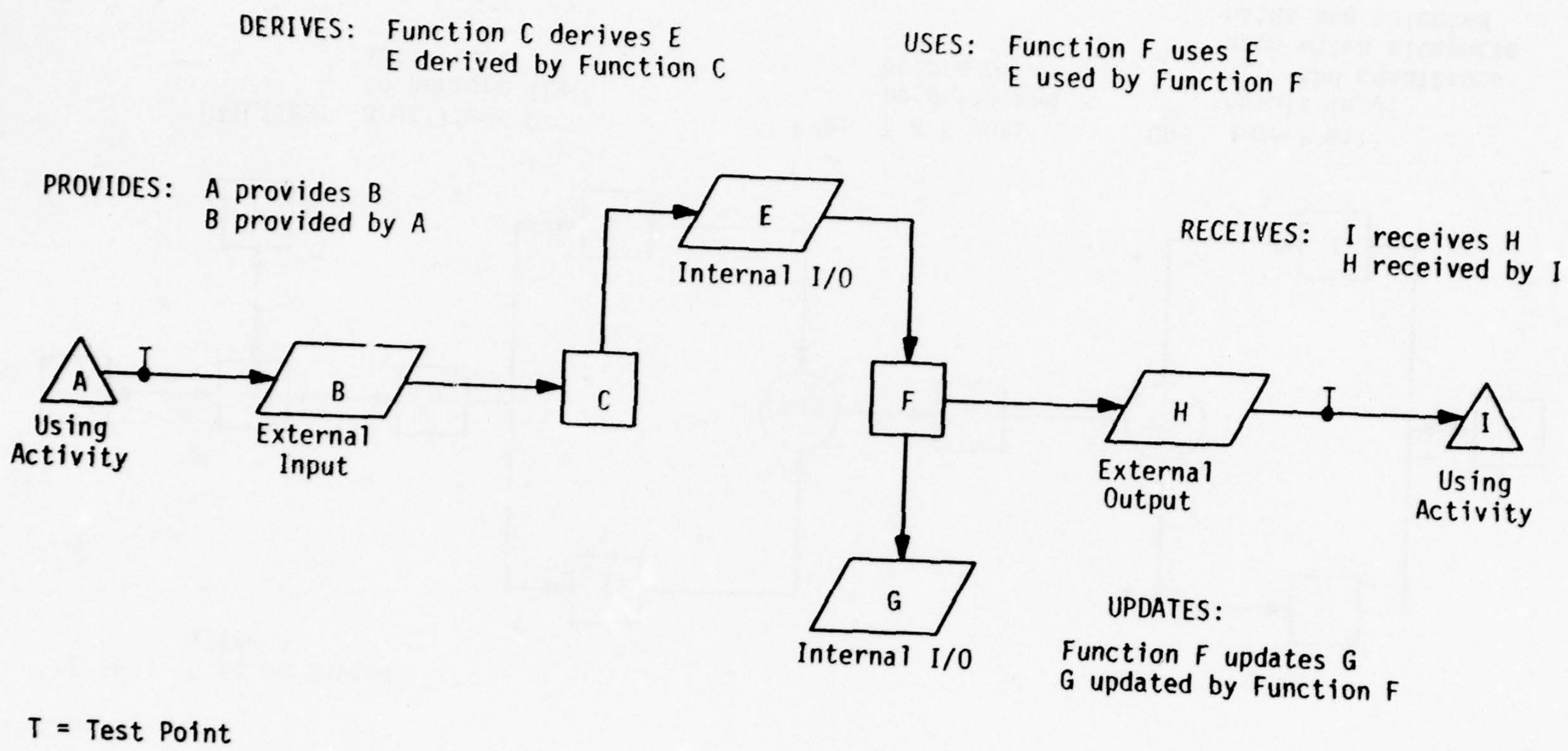


Figure 7. Information-Flow Diagram

SPACE SYSTEM X LOGICON CONTROL FLOW REPORT DATE 01/23/79 Page 1

FLOW OF CONTROL IS FROM LEFT TO RIGHT

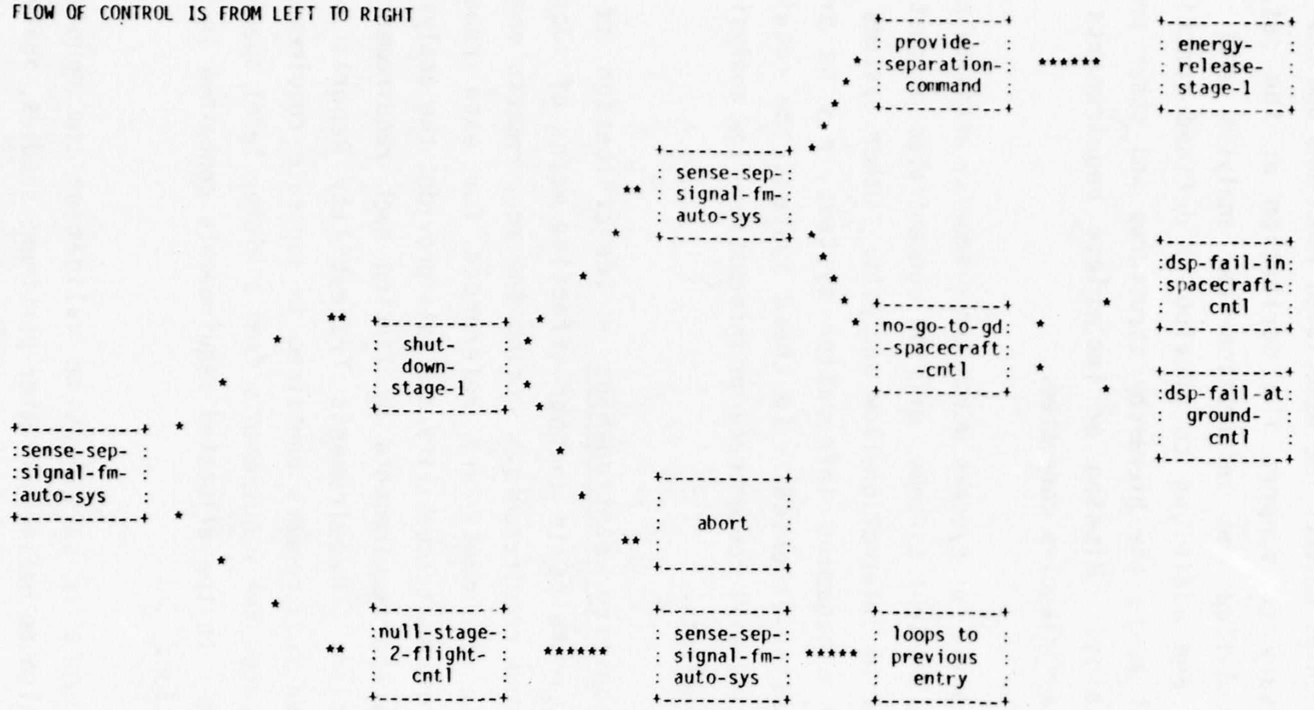


Figure 8. Space System X: Control-Flow Diagram

information-flow analysis is to trace an output back to system inputs, either external data, messages, or stimuli. As a result of this analysis, the various relationships between the associated functions and the internal information necessary to support the derivation of the output are identified. Control-flow and information-flow analysis will identify necessary changes and additions to previously defined functions and constraints as well as to the hierarchy structures and other previously defined relationships. Missing or incomplete requirements can be determined and the deficiencies corrected.

Requirements engineering for systems which are primarily activity oriented, such as command and control systems, will be concentrated on control-flow analysis as opposed to information-flow analysis. Other systems such as communications or management information systems, may be primarily information-processing-oriented. In these systems the requirements engineering activities will concentrate on information-flow analysis rather than control-flow analysis.

4.1.3.3.3 Traceability Relationships - Identification of system traceability relationships is another effective means of identifying incomplete or missing requirements. During the requirements engineering activities, source documents are referenced for each requirement identified. Requirements traceability analysis provides the analyst with a means of verifying the requirements by linking each requirement to all source documentation. Requirements Traceability Reports show the traceability between requirements contained in separate requirements data bases. Figure 9 traces the requirements from a higher level specification of the space-system-x to the allocated requirements contained in the next level of specification.

This form of analysis is pertinent to validating the requirements. Relationships can also be defined to other pertinent studies, analyses, and plans which are being accomplished concurrently with the requirements engineering activities, such as program management directives and plans, system sizing and timing studies, prototyping, simulations, test planning, and the like. System test requirements (quality assurance), as well as

SPACE SYSTEM X LOGICON REQUIREMENTS TRACEABILITY REPORT DATE 01/23/79 PAGE 1

PRIMARY DOCUMENT PARAGRAPH NUMBER	PRIMARY REQUIREMENT NAME	SECONDARY DOCUMENT PARAGRAPH NUMBER	SECONDARY REQUIREMENT NAME
3.7.3.1.4	jettison-stage-1	3.2.3.1	provide-primary-automatic-staging-signal
3.7.3.1.4	jettison-stage-1	3.2.3.4	provide-auto-2nd-staging-signal-at-final-commit-time
3.7.3.1.4	jettison-stage-1	3.2.3.5	provide-auto-2nd-staging-signal-at-programmed-accel
3.7.3.1.4	jettison-stage-1	3.2.3.2	sense-stage-separation-signal-from-auto-systems
3.7.3.1.4	jettison-stage-1	3.2.3.6	no/go-annunciator-to-ground-and-spacecraft-control
3.7.3.1.4	jettison-stage-1	3.2.3.3	display-fail-at-ground-control
3.7.3.1.4	jettison-stage-1	3.2.3.9	ground-remote-manual-signal
3.7.3.1.4	jettison-stage-1	3.2.3.1	display-fail-in-spacecraft-control
3.7.3.1.4	jettison-stage-1	3.2.3.10	spacecraft-remote-manual-signal
3.7.3.1.4	jettison-stage-1	3.2.3.5	read-manual-signal-into-logic-stream
3.7.3.1.4	jettison-stage-1	3.2.3.9	null-stage-2-flight-control
3.7.3.1.4	jettison-stage-1	3.2.3.8	shut-down-stage-1
3.7.3.1.4	jettison-stage-1	3.2.3.2	abort
3.7.3.1.4	jettison-stage-1	3.2.3.6	provide-separation-command
3.7.3.1.4	jettison-stage-1	3.2.3.4	energy-release-and-unfasten-stage-1
3.7.3.1.4	jettison-stage-1	3.2.3.10	jettison-stage-1-from-launch-vehicle
3.7.3.1.4	jettison-stage-1	3.2.3.3	monitor-stage-1-separation
3.7.3.1.4	jettison-stage-1	3.2.3.7	provide-separation-complete-signal
3.7.3.1.4	jettison-stage-1	3.2.3.4	provide-on-command-to-stage-2-flight-control
3.7.3.1.4	jettison-stage-1	3.2.3.2	stage-2-thrust

Figure 9. Space System X: Requirements Traceability Report

subsequent test plans, procedures, and reports, can be effectively related to the system functional requirements. The links to associated system plans, analyses, and studies accomplished prior to, during, and subsequent to the start of formal requirements engineering are crucial to the overall systems engineering concept. The traceability relationships also provide a bridge between requirements engineering activities and subsequent design and implementation of the system, since the requirements can be traced from functional specifications (Type A), development specifications (Type B), to product specifications (Type C); trace relationships can continue into system test activities during the later phases of the system acquisition.

Throughout the requirements engineering activities, the analyst must be able to evaluate the impact of changes to the requirements. Whatever the reason (policy, economics, study or analysis results, engineering change proposals, etc.), the analyst must be in a position to determine the ramifications of changes to the system requirements. Once the area of impact is identified in the requirements engineering products (functional and I/O hierarchies, control and information-flows, etc.), the traceability relationships provide the capability to readily identify associated impacts to the system and to trace the impacts to all other associated documentation such as the program directives, plans, studies and analyses, test plans, associated system specifications (Type A and Type B, etc.) and the like. The impact can be readily analyzed and the appropriate actions taken.

4.1.3.3.4 Summary

Discrete and well-organized requirements support the primary goal of defining the operational mission needs of the using activity while giving the analyst visibility and control over the system definition process. Discrete and well-organized requirements are prerequisites for the creation of good Type A and B specifications.

4.1.4 System Requirement Types

4.1.4.1 Introduction: Understanding the various system requirements types and their use contributes significantly to the identification of discrete requirements and, therefore, quality requirements definitions. System requirements fall into two sets: the functional requirements and the constraint requirements (Table 1). The remainder of this paragraph describes each of these requirement sets, identifies their components, and discusses the relationships between them. The Requirements Engineering Guidebook (Volume III) contains a more extensive description of the system requirement types.

4.1.4.2 Functional Requirements Set: The functional requirements set is the backbone of the system requirements engineering process. It is within this set of requirements that the pure design-free or solution independent needs are declared. Simply stated, the functional requirements represent the total discrete system activities required to achieve a specific objective. A functional requirement identifies what must be accomplished without identifying any aspect concerning the means such as hardware, computer programs, personnel, facilities, or procedural data. The functional requirements represent a problem statement devoid of any overtone or specifics regarding real or conceptual solutions which satisfy any or part of the needed functions. Some examples of discrete top-level functions for an electronic system might be surveillance, tracking, identification, interceptor control, and communications.

4.1.4.3 Constraint Requirements Set: The second set of requirements is the constraint set, which consists of five requirement types: performance, physical, operability, test, and design (Table 1). The constraint set modifies the functional requirements set. Without the constraint set a solution for satisfying the system functional requirements could not be achieved. However, excessive or unrealistic constraints can eliminate all solutions or increase the technical risks and cost of the solution. Therefore, the identification of the constraint requirement set must be achieved with care. Whenever specific constraints are identified, there

Table 1. System Requirement Types

SYSTEM REQUIREMENTS	FUNCTIONAL REQUIREMENTS (functions)	The set of discrete functions which identify the pure design free or solution independent needs of the system as a whole. The functional requirements identify what must be accomplished while avoiding solution statements or overtones.	
	CONSTRAINT REQUIREMENTS (Constraints)	PERFORMANCE	How well the system functions must be accomplished, such as timeliness and accuracy. Also called performance characteristics, MIL-STD-490.
		PHYSICAL	Influences the design solution in a physical manner: power, size, weight, environment, human factors, existing system interfaces, GFP, etc. Also called Physical Characteristics, MIL-STD-490.
		OPERABILITY	Reliability, maintainability, availability, dependability.
		TEST	Identify the functional, performance, physical, operability, and design requirements which will be evaluated during system integration and test.
		DESIGN	The minimum or essential design and construction requirements which are a constraint on the functional requirements of the system during the design and construction of the system end-items (CIs/ CPCIs). Also called Design and Construction, MIL-STD-490.

must be sufficient justification, such as an engineering analysis, which clearly shows that the constraint is reasonable, necessary, and practicable, and represents an actual requirement and not just a desirable feature. The five constraint requirement types are discussed in the following paragraphs.

4.1.4.3.1 Performance Requirements - Performance requirements identify "how well" the functions of the system must be accomplished. These requirements are the essential quantifiable statistical parameters upon which the successful accomplishment of system functions can be evaluated, such as timeliness and accuracy.

4.1.4.3.2 Physical Requirements - Physical requirements constrain or significantly influence the design solution in a physical manner. The physical constraints include power, physical features (size and weight), environmental considerations (controlled or natural), human performance capabilities and limitations (human factors), predetermined internal and external system interfacing, use of existing equipment (off-the-shelf) and Government Furnished Property (GFP), and use of standard parts.

4.1.4.3.3 Operability Requirements - Operability requirements include system availability, and dependability. Availability incorporates the aspects of reliability and maintainability; dependability incorporates the aspects of survivability and vulnerability (S/V), and external electromagnetic interference. Each of these operability categories is influenced by design related issues, policy related impact, or non-controllable factors.

4.1.4.3.4 Test Requirements - Test requirements impact the design process and the resulting system configuration. The test requirements are identified separately from other requirement types to emphasize the importance of the testability of the system requirements.

4.1.4.3.5 Design Requirements - Design requirements represent the minimum or essential design and construction requirements which are not addressed by the previously described constraint requirement types. During the initial phases of systems requirements engineering, certain design and construction standards may be specified directly or by reference to other specifications or standards. As the system development continues, engineering analysis and trade study results (as well as other engineering activities such as prototyping and simulations) may indicate the need for additional design constraints which are practicable and necessary for the system's operation and maintenance (O&M).

4.1.5 Requirements Engineering Procedure

Requirements engineering is an iterative process of defining the system requirements and analyzing the integrity of the requirements for completeness, consistency, testability, and traceability. As the process continues the system requirements are defined and analyzed in progressively expanding manner. The definition and analysis activities will move from one area of concentration to another as the results of previous activities reveal areas needing additional work. No singular approach can be rigidly defined and applied which can take into account the many possibilities which must be considered. However, guidelines for requirements engineering and associated tasks can be defined and then tailored for specific requirements engineering applications. The following is a synopsis of the requirements engineering procedures contained in Volume III, Requirements Engineering Guidebook. The general framework for requirements engineering is illustrated in Figure 10. Each block represents a unique requirements engineering activity which must be accomplished in defining and analyzing system requirements. There is a continual interaction between the activities of each block, and although each block appears as a single activity, it is in fact part of a continuum. Selection of an actual approach for a given application is one of the tasks (BLOCK 2).

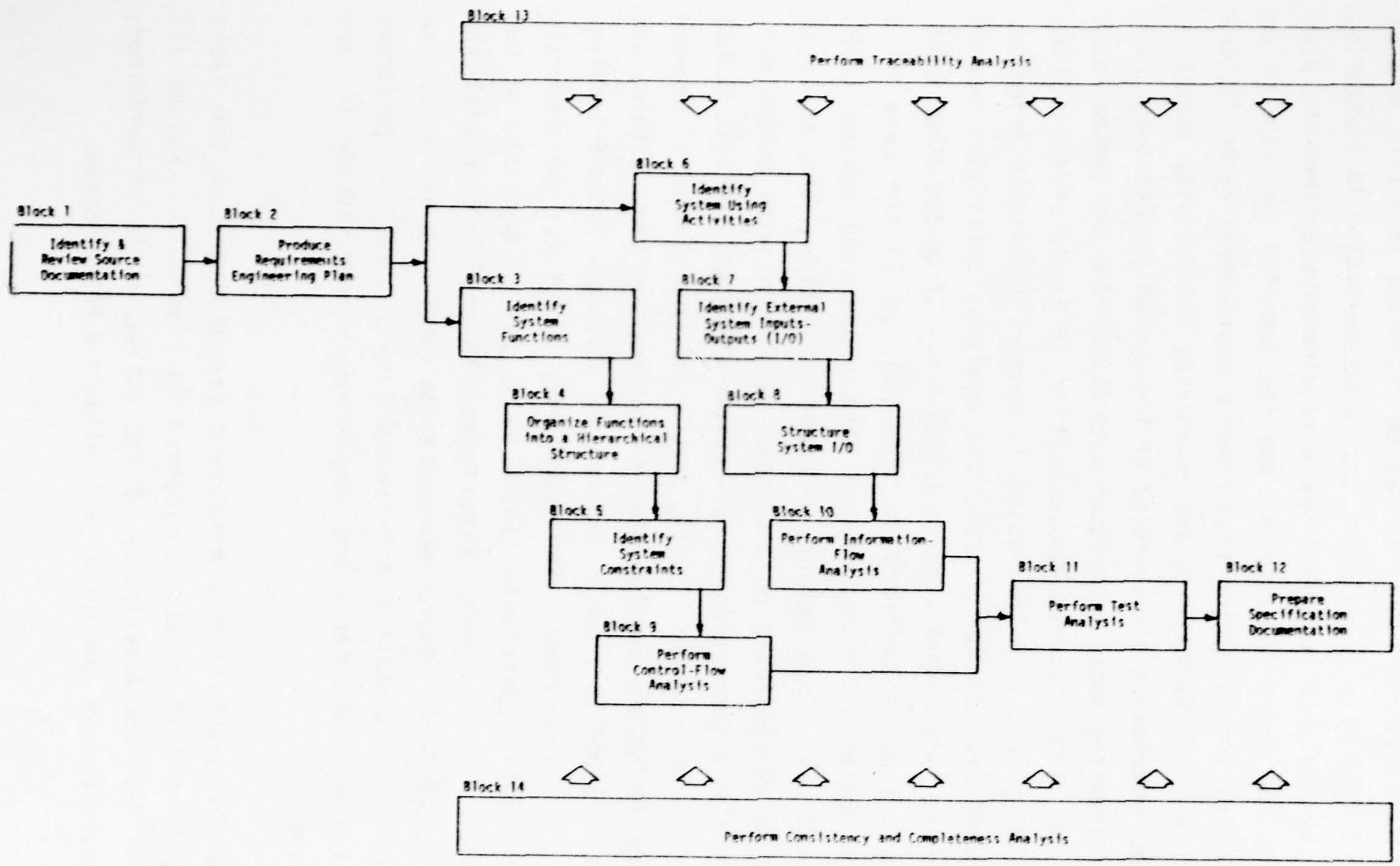


Figure 10. Requirements Engineering Procedures

The activities identified in Figure 10 may be organized into five general steps. In Step 1 (BLOCKS 1-2) pertinent source documentation is identified and reviewed; the analysis team develops a requirements engineering plan which identifies the resources required and the specific approach to be taken in performing the remaining requirements engineering tasks (BLOCKS 3-14). Step 2 involves identifying and organizing the activity structure (BLOCKS 3-5) and information structure(s) of the system (BLOCKS 6-8). The requirements engineering tasks associated with BLOCKS 3-5 are concentrated on analyzing the system source documentation in terms of activities performed by the system. If the system is primarily activity oriented, such as a command and control system, the analysis activities may be concentrated on the tasks identified in BLOCKS 3-5. If on the other hand, the system is primarily information oriented, as in the case of a communications system or an automated data processing system (ADP) application such as a management information system, the analysis activities may be concentrated on the tasks associated with BLOCKS 6-8. Generally the analysis team performs the activities associated with BLOCKS 3-5 and BLOCKS 6-8 concurrently. During Step 3 the flow of control between system functions (BLOCK 9) and the flow of information into, within, and out of the system (BLOCK 10) can be defined and analyzed. Step 4 involves analyzing the system requirements for testability (BLOCK 11) and preparing required specification documents (BLOCK 12). Step 5 consists of two activities which are continuously performed in conjunction with the activities of BLOCKS 3-12. Source documentation references are maintained for each requirement identified and traceability analysis is performed (BLOCK 13). Various consistency and completeness checks (BLOCK 14) are also accomplished.

The following paragraphs are a condensed presentation of the tasks corresponding to the 14 blocks diagrammed in Figure 10. Volume III contains a more detailed description of the activities to be performed, including a description of Conceptual and Validation Phase issues.

4.1.5.1 Identify and Review Source Documentation (BLOCK 1): During this task the requirements analysis team individually reviews the source documentation in order to become familiar with the overall system requirements. The review of the source documentation and the assessment of requirement types are prerequisites for developing the requirements engineering plan (BLOCK 2).

4.1.5.2 Produce Requirements Engineering Plan (BLOCK 2): After review of the source documentation the analysis team determines the specific approach to accomplishing BLOCKS 3-14. This approach takes into account all available resources including personnel, schedule, and financial considerations. The planning includes the methodology to be applied (tools, techniques, conventions, etc.) specific tasks to be accomplished, personnel assignments, resource descriptions, schedules and milestones, preliminary and final documentation to be produced (BLOCK 12), progress reviews and quality assurance procedures. If automated tools are selected to assist in the requirements definition and analysis of the source documentation, features of tools to be employed are determined. The results are described in a requirements engineering plan.

4.1.5.3 Identify System Functions (BLOCK 3): During this task the source documentation is analyzed and the system functions, necessary to control or produce the desired outputs from the available inputs, are identified. A function is a discrete activity within a system. The collection of discrete functions, defines the total activities which must be accomplished by the system to achieve a given objective. The functions identified range from high level (first possible functional breakout of the system) to detailed lower level functions which represent finite, distinct actions to be performed by system equipment, computer programs, personnel, facilities, procedural data, or combinations thereof. Each function is given a unique name conforming to the function name in the source documentation or its characteristics. The requirements definition and analysis activities associated with this task are oriented toward identifying the actual user functional requirements which are necessary to achieve the mission objective. As each function is identified and named,

the primary and secondary references to the source documentation are maintained (BLOCK 13). Each function is supplemented by a description of the function and its purpose, a statement of the conditions under which the function is activated, and a description of the system external and internal inputs/outputs that the function will receive, use, or generate. The latter descriptions serve as a basis upon which the requirements engineering activities of BLOCKS 7, 9, and 10 will proceed.

4.1.5.4 Organize Functions into a Hierarchical Structure (BLOCK 4) - In conjunction with identifying the system functions as described in BLOCK 3, the functions are arranged into logical hierarchical structure diagrams (Figure 3). This form of organization is suited for structuring system functional requirements in a logical arrangement for communicating system functions and the relationships between the functions to design engineers. This form of organization provides a view of the system as an aggregate of functions broken out into a logical arrangement of subordinate discrete activities which must be performed. This logical form of organization is distinguished from the control-flow (BLOCK 9) and information-flow (BLOCK 10) forms of organizing system functions. The functions of the system are grouped into higher levels of organiztion representing the first possible breakout of the system. Upper-level functions are refined by the identification of subordinate levels.

Each level of the hierarchy is limited to six functions or less. The desire to limit the number of functions at each level is similar to the need to limit the number of lines of code in a computer program. In the latter case the problem is not that the computer can not handle large programs, but that the understandability of the program is enhanced in the human sense. The same is true for the hierarchical breakout of the system functions. Experience and study has shown that a limit of six is helpful [18]. The object of the functional hierarchical presentation is to provide a greater human understanding of system functions.

In a functional hierarchy the sum of the activities of the functions on a given level is equal to the activity at the next higher level in the hierarchy. This principle means the total system activities are defined

by the functions at the lowest level in the hierarchy.

4.1.5.5 Identify System Constraints (BLOCK 5): In conjunction with the identification of system functions and organizing functions into a hierarchical structure, the analysis team will identify all system constraints. The constraint requirements are limited to system performance, physical, operability, design. Test Requirement constraints are addressed under BLOCK 11. Constraint requirements are derived from available source documentation or from the results of trade-off studies, feasibility studies or advanced development studies. Constraints which are not clearly justified from available documentation are eliminated from consideration until documented justification is available. All constraint requirements are stated in specific quantifiable parameters, either as a single value or range of values, including the unit of measure, limits, accuracy or precision, and frequency.

4.1.5.6 Identify System Using Activities (BLOCK 6): Using activities (organizations, operational units, or operator positions) which interact with the system are identified. The identification of using activities provides the basis of information-flow analysis (BLOCK 10). The identification includes the names of using organizations identified in the source documentation or through other determinations such as human engineering studies. Lower level position names, such as specific operator positions are identified and described to the level of detail required for the associated functions.

Using activities are a form of design constraint but are separately identified to support other requirements engineering activities such as information-flow analysis. Whenever using activities are identified, there must be sufficient justification, such as engineering analysis, which clearly shows that the using activity is necessary and represents an absolute requirement and not just a desirable feature.

4.1.5.7 Identify External System Inputs-Outputs (BLOCK 7): In conjunction with identifying the using activities, the analysis team will identify the output (responses) required from the system. Output

information consists of system messages and reports necessary for the operation, maintenance, control of the system, and support of the mission objectives. Subsequent to each output being defined, the associated system inputs (stimuli) are identified. The input information may be used directly from the external source or used by the system (see BLOCK 10) to derive all or part of an output. Inputs and outputs are associated with their respective sources or destinations. These sources and destinations may be the using activities or external systems. Additional informational requirements, such as internal information necessary for the system's operation, are identified during BLOCK 10.

4.1.5.8 Structure System Inputs-Outputs (BLOCK 8): Concurrent with BLOCK 6 and 7 activities, the system inputs and outputs (I/O) are arranged into hiearchical structure diagrams (Figure 5). The emphasis on the I/O hierarchical structures is to organize the I/O and their subordinate parts into logical organizations or simply into groupings of information. Structuring the I/O is an effective means of identifying incomplete or missing I/O requirements and for communicating the input and output requirements to design engineers. Parts of I/O identified during BLOCK 7 are associated with other I/O and organized into hierarchical structures. Changes and additions to the I/O hierarchical structures may be required as information-flow analysis (BLOCK 10) is accomplished. The upper parts of the individual I/O hierarchical structures are equivalent to the aggregate of the subordinate parts in the hierarchy.

4.1.5.9 Perform Control-Flow Analysis (BLOCK 9): After the functions of the system are identified (BLOCK 3), the control flow between the functions is described in control-flow diagrams. Control-flow analysis provides a means of viewing the system from an activity-oriented perspective and is often referred to as functional-flow analysis. The control flow diagram (Figure 6) describes the sequential flow between the system functions. The control flow diagrams indicate only the relationship between system functions and does not imply any lapse in time or intermediate activity. Conditions which determine the flow directions are described using the following control-flow relationships as illustrated in Figure 6.

SERIES This is a sequential relationship between two or more activities. This relationship is assumed unless an AND, OR, or UTILIZE relationship is indicated in the flow path.

AND Activities preceding the AND must be accomplished before the flow may continue.

OR Any one of the alternate paths may lead to the next activity. The conditions upon which the alternate paths are selected are associated with the OR.

UTILIZES This relationship indicates that a function on a path is dependent upon the use of one or more other functions in order to accomplish its activities. A single function or sequence of functions may be defined once and utilized as frequently as necessary in the control flow without having to be redefined (replicated) for each use.

4.1.5.10 Perform Information-Flow Analysis (BLOCK 10): This activity builds upon the I/O hierarchical structure (BLOCK 8) by providing a means of analyzing the system as an information processing system (Figure 7). During this analysis, the flow relationships between external system inputs and resulting outputs are identified in information-flow diagrams. These diagrams provide the basis for determining that each I/O is used, derived, or updated. An effective method of information-flow analysis is to trace an output back to the system input: external data, messages, or stimuli. This method permits the relationships between associated functions and the internal information necessary to support the derivation of the output to be identified. The flow associations between system information are described using the following information-flow relationships as illustrated in Figure 7:

USES This relationship indicates that a function on the path uses external information (external input) or internal system information (internal input) in order to accomplish its activities.

DERIVES — This relationship indicates that a function on the path derives either external information (external output) or internal system information (internal output) as part of its activities.

UPDATES — This relationship indicates that a function on the path updates internal system information as part of its activities.

The informational flow indicates the relationship between system functions and system information (external and internal system I/O) and does not imply any lapse in time or intermediate I/O being used, derived, or updated. These relationships are identified for each level in the information hierarchy. As the information analysis continues, the relationships are allocated to lower levels in the information hierarchy as the I/O is identified (BLOCK 7) and structured (BLOCK 8).

For the purpose of information-flow analysis, the using activities identified during BLOCK 6 are integral to the definition of the system as an aggregate of hardware, computer programs, personnel, facilities, and procedural data. The relationships between the using activities are described using the following information-flow relationships as illustrated in Figure 7:

PROVIDES — This relationship indicates that a using activity is the source of the external input.

RECEIVES — This relationship indicates that a using activity is the recipient of the external output.

4.1.5.11 Perform Test Analysis (BLOCK 11): Test requirements identify the system requirements which will be evaluated during system integration and test. The principle objective of test analysis is to identify which areas in the system definition shall undergo formal test and verification. This is achieved by identifying test points on the control-flow and information-flow paths (Figures 6 and 7). As the control flows and

information flows evolve, the analysis team determines test points on the flow paths. These test points are added to the flow paths at the selected test data sampling locations. The selection of test points is accomplished concurrent with the test planning activities. As test cases are determined by analysis of the control and information flows, the test points are described and associated with test plans and procedures.

The association between system test plans, analyses, and studies documented prior to, during, and subsequent to the start of formal requirements engineering is crucial to the overall requirements engineering concept. Documented test objectives preceding formal requirements engineering are analyzed. As a result, test points in the control and information flows are selected which provide data for various test cases which support testing objectives.

4.1.5.12 Prepare Specification Documentation (BLOCK 12): The preparation of specification documents is accomplished in accordance with MIL-STD-490 as supplemented by MIL-STD-483 (USAF). Specifications serve to document the system requirements throughout the system acquisition life cycle. In Air Force acquisitions these documents are an integral part of the management concept: configuration management, data management, system integration and testing, and contracting. The system requirements definition and analysis activities (BLOCKS 3-11) provide the basis upon which the preparation of specification documents proceeds. The products of BLOCKS 3-11 (functional hierarchical structures, I/O hierarchical structures, control flows, and information-flows, etc.) are incorporated directly into the specification documents in accordance with the prescribed format of MIL-STD-490/483. All requirements in the specification documents are traceable to the products of the requirements engineering performed as described in BLOCKS 3-11. Therefore, each specification document is cross-referenced to the requirements engineering products (BLOCKS 3-11).

Volume III (Tables 2 and 3) provides a cross-reference between the requirements engineering activities described in the guidebook and the associated paragraph requirements in MIL-STD-490/483 (USAF) for Type A, system/segment specifications and for Type B5 computer program configuration item specifications.

4.1.5.13 Perform Traceability Analysis (BLOCK 13): System requirements traceability is another effective means of identifying incomplete or missing requirements. Traceability gives the analyst a means of verifying the requirements by linking each requirement to the varying forms of source documentation such as program directives and plans, studies, analyses, test plans, associated specifications (Type A, B, etc.) and the like. Throughout the requirements engineering activities (BLOCKS 3-11), each requirement is associated with the sources of the requirement (source documents). Two forms of references are provided: primary and secondary source references. Primary source references refer to specific paragraphs in source documentation which are the origin of the requirement. Secondary source references refer to specific paragraphs in the source documentation which provide information about closely related requirements, discussions of the rationale about the requirement, or other useful background information.

4.1.5.14 Perform Consistency and Completeness Analysis (BLOCK 14): Throughout the requirements engineering activities (BLOCKS 3-13), analysis of the consistency and completeness of the requirements definition assures the integrity of the system being defined. Associated with each requirements engineering activity are various consistency and completeness checks which are performed concurrent with each block.

4.2 Requirements Engineering Tool Capabilities

An abbreviated list of the automated tool capabilities which support the activities described in the Requirements Engineering Guidebook (Volume III) is provided in Appendix D. The following paragraphs describe the role of automation in requirements engineering and summarize the automated tool capabilities presented in Appendix D.

4.2.1 Intrinsic Capabilities of Automated Tools

Automated tools like CADSAT assist requirements engineering four ways:

- Provide a medium for formal requirements definition
- Perform rudimentary analysis
- Produce documentation
- Permit a flexible, iterative approach to requirements definition

Automated tools like CADSAT incorporate a language which permits formal definition of essential aspects of the requirements. The computer can manipulate the information described in this language to help the analyst perform necessary tasks. The computer can serve a search and retrieval function. Automated tools can readily produce timely documentation. Because the computer stores information electronically, changes are made easily. This capability allows the analyst to take an iterative approach to requirements definition. The requirements may be defined by making successive refinements.

Automated tools do not directly analyze the source requirements. Automated tools analyze, in a rudimentary, non-creative way, information about requirements which have been recorded in a specific language and entered into a computer (in a requirements data base). A successful automated tool must help the analyst assure that the information in the requirements data base does indeed reflect the system requirements.

4.2.2 The Automated Tool

Automated tools like CADSAT consist of two parts, a language and an analyzer. The language provides the means for describing the requirements for functional and development specifications[1]. The report language and

[1] In Air Force system acquisitions the functional specification is the system/segment specification (Type A, MIL-STD-483 (USAF), Appendix III) and the development specifications are Type B specifications. The Computer Program Configuration Item Specification (Type B5, MIL-STD-483 (USAF), Appendix VI) is the primary development specification addressed in this study.

analyzer can be used by the analyst in completing the tasks described in the Requirements Engineering Guidebook (Volume III). The tool should aid in the formalization of the requirements in the requirements data base and help to ensure that all of the requirements in the source material appear accurately in the requirements data base. The tool should help the analyst assure the completeness and consistency of the requirements in the requirements data base. The tool should produce the necessary system documentation.

4.2.2.1 The Language

The language objects and relationships described in the following paragraphs incorporate all the system requirements and provide the means to analyze the requirements through automated means.

4.2.2.1.1 Language Objects - The "nouns" of a requirements definition language are called objects. For example, there are objects for describing system functions and other objects for describing the various kinds of constraint requirements: performance, physical, operability, design and test. The source document reference is another object. The tracekey permits requirements from one set of documentation (represented by one requirements data base) to be linked to another set (represented in a second requirements data base). Tracekeys are either source references or unique identifiers for the requirements in the second requirements data base. The analyzer generated unique requirement identifier is also an object.

Other objects describe system external and internal inputs and outputs (I/O). These permit the definition of both physical and logical I/O structures. The physical data structures are formal associations of I/O which are necessary to describe the system (Track numbers are always associated with certain other tracking data, for example.) or are predetermined external constraints. (A specific message structure might be mandated by circumstances.) A logical I/O structure is simply a collection of inputs and outputs associated only for the convenience of the analyst.

Each object is named. For instance, the requirement, "sense stage separation signal from automatic systems," is a functional requirement and might be entered in the requirements data base as a function object called

sense-stage-separation-signal-from-auto-systems.

The few objects described above are sufficient to formalize the requirements and will permit extensive automated analysis. Depending on the application of the automated tool, not every aspect of the requirements has to be formalized. The essential objective is to make the requirements discrete and to organize the requirements as a basis for further analysis.

4.2.2.1.2 Language Relationships - The language should allow various relationships between objects to be described. These are the "verbs" of the language.

Several relationships describe requirement-to-requirement and requirement-to-document associations. For example, relationships describe the hierarchical structure of functional requirements (Figures 3 and 4). For example, tracking control and tracking calculations might be considered to be subfunctions of tracking. Another relation associates constraints to functions. A single constraint can be associated with more than one functional requirement. The Related-Requirement relationship associates any two requirements that the analyst deems associated in any manner. Every section of the source material which discusses a requirement, no matter how indirectly, should be associated with that requirement as a source reference. These source relationships are used later by the analyzer for traceability, and consistency and completeness analysis. Since the association of many source references with a single requirement can cause confusion as to the nature of the requirement, two types of source relationships are employed: a primary and secondary source relationship. The primary relation is employed to associate the most comprehensive source references with the requirement.

Another relationship establishes the hierarchical structure of system external and internal inputs and outputs (Figure 5). This relationship enables the definition of both physical and logical I/O structures.

Some relationships describe the flow of information into, within, and out of the system (Figure 7):

- uses/used by,
- derives/derived,
- updates/updated,
- utilizes/utilized by
- provides/provided by
- receives/received by

The "uses" relationship indicates that a function on the path uses information external to the system (external input) or internal system information (internal input) in order to accomplish its activities. The "derives" relationship indicates that a function on the path derives (generates) either information external to the system (external output) or internal system information (internal output) as part of its activities. The "updates" relationship indicates that a function on the path updates internal system information as part of its activities. The "utilizes" relationship indicates that a function on the path is dependent upon the use of one or more other functions in order to accomplish its activities. A single function or sequence of functions may be defined once and utilized as frequently as necessary in the flow path without having to be redefined (replicated) for each use. These information flow relationships are illustrated in Figure 7. The "provides" relationship indicates that a using activity (organization, operational unit, or position) is the source of the external input. The "receives" relationship indicates a using activity is the recipient of the external output.

Other relationships describe the flow of control among functions (Figures 6 and 8). The "trigger" relationship describes other functions which follow in the sequence of system flow. The "conditional trigger" describes other functions which, under certain boolean conditions (AND, OR), follow in the

sequence of system flow. The "utilizes" relationship simplifies the description of control flow when a single function appears many times in the control-flow sequence.

Each requirement object (function, constraint, I/O, etc.) and relationship (functional and information (I/O) hierarchy, control and information flow, etc.) can be supplemented by an ordinary English description. The description may be a complete statement of the requirements which the object or relationships represent. The text could also be a synopsis of the pertinent aspects while the source reference provides a more complete description.

4.2.2.2 The Analyzer

The analyzer is the second part of an automated tool like CADSAT. It generates a series of reports. The reports, essential for the application of the Requirements Engineering Guidebook, can be grouped into six general report categories: requirements data base management, functional analysis, I/O analysis, traceability analysis, test analysis, and documentation.

4.2.2.2.1 Requirements Data Base Management

Change Requirements Data Base Reports - The analyzer handles requirements definition entries into the requirements data base and changes definitions already in the requirements data base. The report is in the form of a display of changes made to the requirements data base. New requirements definitions are entered into the requirements data base with the option of updating or not updating the existing requirements data base (Input report), existing objects or relationships are deleted (Delete Objects, Delete Relationships reports), objects are renamed (Rename Objects report) or the object types changed (Change Object Type report). The analyzer checks all additions or changes to the requirements data base for consistency with the rules of the language, and any errors are noted on appropriate reports. To help the analyst track his work, all entries and changes to the requirements data base are also recorded on appropriate

printouts. The Input Report has an update feature to precheck the legality of requirement data base entries. All new source references are checked by the analyzer for consistency against complete source references (lists of paragraph numbers, etc.) and any errors are printed. This feature helps to prevent incorrect source references. As an option, the operator can specify that the analyzer note when cases of multiple references to the same source paragraphs are encountered. Under these circumstances, the analyzer names and lists source references for the requirements with common source references. The analyst uses this information to find source reference errors, avoid duplication of requirements, and find requirement overlaps.

Object Information Report - The object information report is used to check the contents of the requirements data base. Provided with a list of object names, the report supplies a list of any or all information about each object. For example, given the object name "tracking" and asked to provide functional hierarchy and source information, the analyzer might return: superior function is "surveillance", subfunctions are "track control", and "track calculations" and source is 3.7.1.7.4.3. Given the object name "tracking" and asked to provide all information, the analyzer might return the information given above plus any other information in the requirements data base concerning "tracking". The output of this report would have the same format as the Input and Delete Relationship reports described above (4.2.2.2.1). With a common format to these change reports, the Object Information report can be output and used for updating the requirements data base. This procedure minimizes some of the clerical work associated with requirements data base changes.

Source Document Summary Report - The Source Document Summary report is used to compare the requirements data base contents against the source documentation. The report presents a sequential list of all source document references. The requirements associated with each source reference are printed adjacent to each source reference. The nature of the source of the requirement relationship is noted (primary or secondary).

When no source references have been associated with a requirement, no reference appears adjacent to the requirement. The report helps to find source document requirements which have not been incorporated in the requirements data base. Comparing this report with the source document, the analyst can assure that the requirements data base completely reflects the documentation.

Identify Specified Objects Report - The purpose of this report is to retrieve requested object and relationship information from the requirements data base. It relieves the analyst of simple but time consuming tasks. For example, this report aids finding sources which have not been referenced or functions which have no control or information flow relationships. The report is a list of object names which may be used directly or used to determine the need for additional reports such as the Object Information report.

Requirements Data Base Status Reports - The requirements data base status reports provide summary information on the contents of the requirements data base. Requirements data base objects are listed along with various statistics showing the quantities, percentages (as appropriate), and quality of each object in the requirements data base. Statistics presented for each object might include the status of the requirements (complete, incomplete, etc.), relationships between requirements (sources/tracekeys, triggers, uses/derives/updates, etc.), and number of lines of descriptive text in the requirements data base.

4.2.2.2.2 Functional Analysis

Functional Hierarchical Structure Report - The primary purpose of this report is to provide requirements visibility. The report uses the functional hierarchical structure information contained in the requirements data base to present the breakdown of system functions from the general to the specific. Using this report, the analyst can readily grasp how a single function fits into the overall system. By following the hierarchy from top to bottom the analyst can also use the report to locate system requirements.

The secondary purpose of the Functional Hierarchical Structure report is to present requirements data base information in a format that is easily used by the analyst. The report presents, in association with selected functional requirements, any or all of the following: sources, constraints, system information and control flow relationships. The analyst can use the report with options to check the status of the requirements data base. The hierarchical structure format gives the requirements data base information a useful and convenient context. This format is more useful, for example, than a presentation of requirements data base material via an alphabetical listing of objects or requirements.

Control Flow Report - The Control Flow report helps identify the completeness and consistency of system control flow. On input of a function name, the report traces the control flow forward or backward by a specified number of functions. The report provides an overview of any portion of the control flow for critical examination by the analyst. Missing control flow logic is highlighted by a premature termination of the flow sequence in the report.

I/O-Function Interaction Report - The I/O-Function Interaction report shows the information flow for selected functions or I/O. The report is useful when the analyst is concerned with a portion of the system relative to a selected group of I/O. It answers such questions as "How does the system I/O tie these functions together?" or "Where does this I/O fit into the system?"

4.2.2.2.3 I/O Analysis

I/O Hierarchical Structure Report - This report prints selected parts of the I/O structure. The report is used by the analyst to review and upgrade the I/O structure. The final results provide visibility into the system I/O structures.

Information-Flow Report - Information-Flow reports help to assure a complete and consistent I/O description of the system. Given a system

input or output name, the report illustrates the sequence of functions and associated I/O which are needed to derive the I/O. The report can also be prompted to trace system I/O from the system external inputs toward the external outputs (or vice versa).

The report helps the analyst to examine the information flow for logical errors and inconsistencies. When the report is unable to trace back to system external inputs, missing functions or flow relationships are indicated.

The report is different from the I/O-Function Interaction report in that it concentrates on the evolution of a single input or output rather than the I/O interactions among a group of selected functions or I/O.

This report is also useful for change impact analysis. The report identifies requirements affected by changes in system inputs and outputs identified. The report further leads the analyst to the impacts of any requirement changes affecting the system I/O.

4.2.2.2.4 Traceability Analysis

Find Related Requirements Report - This report aids change impact analysis by using the requirements data base information to locate requirements which are in some way related to a requirement which may be changed. The analyst specifies the combination of relationships employed to find the related requirements. The report is also useful to help find and resolve requirement inconsistencies or redundancies.

Requirements Traceability Report - The Requirements Traceability report shows the traceability of requirements from one set of documentation to another. Various options are provided. Requirements are traced from the source documentation to the requirements data base, or from the requirements data base to a second requirements data base (based on a second set of source documentation) or from the source documentation to the

second requirements data base. The tracekey object in one requirements data base matches a unique identifier or source reference (depending on option) to provide the link between requirements data bases. The report provides a convenient summary of requirements traceability information.

4.2.2.2.5 Test Analysis

Test Reports - The test reports are used to evaluate the quality and completeness of test plans and procedures. Reports can be prepared for each test case defined for the information and control flows. The test reports show the relationship between system flow and associated test cases, test plans and procedures and other pertinent source documentation.

4.2.2.2.6 Documentation

Requirements Document Reports - The requirements document reports are automated reports which can be used directly in system documentation. These reports should conform to the format requirements of the prescribed documentation standard, such as MIL-STD-490/483 (USAF).

The automated requirements engineering tool should provide the capability to identify requirements by a unique requirement number. This capability allows the analyst to derive a set of unique requirement numbers which aids in refering to the requirements during the requirements engineering activities. In addition, various automated reports (listings and graphic figures generated from the requirements data base) can be easily cross referenced to the requirements text in the resulting system documentation.

At a minimum, a report should be provided where the text of the requirements are presented in the sequence of (and adjacent to) the unique identifiers. This capability provides the analyst with a simple report which is useful during the requirements engineering activities and in preparation of final system documentation. This text report should also include source references and tracekeys which show where the requirement originated and where they are reflected in subsequent documents.

4.3 Requirements Engineering Example

4.3.1 Introduction

An example of the use of the Requirements Engineering Guidebook (Volume III) has been prepared and included in Appendix E of this volume. The example has been derived from the actual requirements engineering activities associated with an Air Force surveillance system acquisition. Excerpts from the surveillance system segment specification (Type A) have been included at the end of Appendix E. The example presents a description of the actual requirements engineering performed on the specification in conjunction with the use of an automated requirements tool, Logicon-Extended CADSAT.

An automated requirements engineering tool like CADSAT helps the system engineer define a set of discrete system requirements. Discrete requirements are desired because they are non-redundant, testable, traceable, and communicable and are amenable to tests for completeness and consistency. Once the requirements have been tentatively defined, the tool helps the system engineer analyze for completeness and consistency. The tool can generate specification documents.

4.3.2 Automated Tool Capabilities

Using a requirements engineering tool like CADSAT, the system engineer incorporates in a requirements data base certain salient features of each system requirement. The requirement is given a descriptive name. References to descriptions of the requirement are associated with each requirement (primary source). References to supplementary or related material may also be associated with the requirement (secondary source). The requirement is categorized as a functional requirement or one of several constraint requirements. A functional hierarchy is developed. Constraint requirements are associated with functional requirements. The flow of control (processing sequence) is defined. System information flows (I/O used and derived by system functions) are described. Finally, a comprehensive description of the requirement is incorporated in the requirements data base.

The automated tool allows the various system requirements to be incorporated in the requirements data base when they become known to the engineer. This is a prime advantage of an automated tool - the file of information about a system requirement is built up in an evolutionary manner. The requirements data base development proceeds in the same way that the system engineer naturally develops an understanding of the system. For example, when an additional reference to a previously identified requirement is found, this new reference is added to the requirements data base. If, during the course of an information-flow analysis, the engineer finds that he previously misunderstood the requirement, or the specification developer receives feedback from the intended system users, appropriate corrections can be made to the requirements data base.

As the requirements analysis proceeds, the automated tool generates intermediate reports which help the systems engineer to complete and refine the requirements definition. These reports present, in various formats, the information which has been incorporated in the requirements data base. The reports present the results of various kinds of analysis. For example, one report lists selected requirements with associated source references or other requirements data base entries in a top-down hierarchical manner. Another report shows how the source documentation has been decomposed into requirements and points to those areas which have not been considered. Other reports show the control and information flows.

An automated tool like CADSAT can generate reports which provide high requirement visibility and provide final system documentation.

4.3.3 Example Requirements Engineering Procedure

Appendix E presents a detailed discussion of the requirements engineering activities performed for an excerpt of the surveillance system segment specification. This example illustrates the requirements engineering activities of the two stages described above including definition and analysis steps and selected automated tool example reports. This example is cross referenced to the requirements engineering activities (BLOCKS

1-14) of the Requirements Engineering Guidebook (Volume III). The following paragraphs are repeated in Appendix E to make the example more comprehensive for inclusion in the RADC Computer Software Development Specification (CP078779-6100D).

Although a tool user could define all aspects of a requirement when that requirement is first identified, experience has shown that it is convenient to proceed in two stages. During the first stage the basics are defined. The requirements are identified, characterized and the functional hierarchy described. The control and system information flows are described during a second stage. This separation into phases is not hard and fast, and appropriate changes and additions are made at any time. The two-stage approach was applied in the example.

4.3.3.1 Stage One: The first stage of the requirements data base development in the example (Appendix E) consists of identifying the basic system functions, primary and secondary source references, organizing the functions into a functional hierarchy and identifying constraint requirements.

4.3.3.1.1 Identification of Functional Requirements (BLOCK 3) - One of the first steps in the development of a requirements data base is the identification of functional requirements (BLOCK 3). Each functional requirement is given a descriptive name. All appropriate source references are associated with the requirement (BLOCK 13). A complete statement of a functional requirement includes a description of the required function, a statement of when and under what conditions the function is activated, and a description of the system I/O that is to be transferred and the results. Source material frequently describes part of a requirement in one place and part in another. Sometimes minor variations in the activities of the function are discussed separately from the fundamental function. The conditions under which the activities of the function are to be performed may be discussed in the source material separate from the discussion of the function. All of this information about the requirement is consolidated in the requirements data base. References to source material which are necessary to completely describe the requirement are called primary references.

4.3.3.1.2 Identification of Secondary Source References (BLOCK 13) - Secondary source references are described along with the primary references.[1] The secondary references refer to information about closely related requirements, discussions of the rationale for the function or any other type of useful background information. This is information which will be reviewed when the requirement is incorporated into a specification (BLOCK 12) and might also be employed later to aid change impact analysis.

4.3.3.1.3 Development of a Functional Hierarchical Structure (BLOCK 4) - The functional requirements are organized into a hierarchical structure (BLOCK 4). This structure shows how the functions break down from the general to the specific. Source references to descriptions of general functional capabilities are associated with the appropriate level in the hierarchy (BLOCK 13). Dummy headings (with no source reference) may be created where necessary to group a set of related requirements logically. The functional breakdown continues until a complete understanding of the preceding higher level function is achieved.

4.3.3.1.4 Identification of Constraint Requirements (BLOCK 5) - Constraint type requirements and their association with the appropriate functional requirements (BLOCK 5) are identified. The constraint requirements include performance, physical, operability, test and design requirements. Source references are associated with constraints in the same manner as with functions.

4.3.3.2 Stage Two: The second stage of the requirements data base development in the Appendix E example consists of the formal description of system information (BLOCK 10) and control flows (BLOCK 9). This stage

[1] The concept of secondary references and other concepts presented in the Appendix E example are a combination of current and proposed CADSAT capabilities resulting from this study.

of the requirements data base development is normally performed after stage one is complete for the whole system. It has been found easier to perform the stage two analysis after the functional and constraint requirements have been tentatively identified and source references associated with the requirements. Stage two activities consist of the identification of system external inputs and outputs, the definition of information flow and control flow.

4.3.3.2.1 Identification of System External Inputs and Outputs (BLOCK 7)
Much of the I/O that is to be manipulated by the system can be identified by examining the information which is input to and output from the system. Input and output messages, operator inputs, and displays should be examined to identify the I/O handled by the system. Other system I/O which is known to the analyst from the stage one analysis should be formally identified and entered into the requirements data base at this time. Each input and output is given a short descriptive name. For communicability, the names should closely resemble those used in the source documentation. Not all system I/O will be recognized at this point. Additional I/O will be included at a later stage in the analysis.

A structure is provided for the I/O (BLOCK 8). There are two types of I/O structures: physical and logical. An example of a physical data structure is a message. A message is a collection of I/O which has a definite physically recognizable structure. In the example (Appendix E), physical collections of information are called "entities". A logical I/O structure is simply a convenient collection of information. For example, one might wish to collect all kinds of weather information under the heading weather data. The various kinds of weather information need not have any physical relationship. In the example (Appendix E) organizational collections of I/O are called "sets".

4.3.3.2.2 Definition of Information Flow (BLOCK 10) - The system engineer considers each function to define the inputs received or used, and the outputs derived by that function. To aid this process the engineer employs

the I/O hierarchies generated (BLOCK 8) and the functional hierarchy (BLOCK 4) constructed during the first stage of the requirements data base development as previously described. Since the I/O identification is probably not complete, some new I/O may be identified during information flow analysis. If the I/O used and derived by the function is not clearly described in the source documentation, then the requirement is not completely defined.

4.3.3.2.3 <u>Definition of Control Flow (BLOCK 9)</u> - The control flow shows the sequence of system processing. This sequence is specified in two ways. A "conditional trigger" is used to indicate a function that sometimes follows. The "trigger" specifies processing that necessarily follows. If the control flows cannot be defined from available source documentation, then the requirements are not properly defined.

The definition of control flows may be facilitated by making use of the "utilizes" relationship. A primary function is said to "utilize" a secondary function when the secondary function supplies a result which is necessary for the operation of the primary function. The "utilizes" relationship is employed to simplify the illustration of control flows when the same function is required many times under different circumstances.

4.4 Implementation Approach

4.4.1 Overview

The requirements engineering procedures for defining and analyzing system requirements in the early phases of the Air Force system acquisition life cycle are presented in a separate volume of this report (Volume III). This separate presentation was selected to provide a complete and concise document which is more readily usable for requirements engineering purposes by Air Force program offices and for potential contracting purposes. The implementation of the Requirements Engineering Guidebook must address certain issues, practices, and policies within the Air Force systems acquisition life cycle.

As discussed previously (Section 3), the procedural requirements of AFSCM 375-5 have not been translated into current Air Force regulations, standards, or specifications. There have been varying approaches to requirements engineering within Air Force program offices and contractor engineering organizations. This shift toward a more laissez-faire systems engineering concept has left the Air Force program offices with only a minimum understanding of what to expect, since there is essentially no basis for evaluation. New engineers entering the program office without formal guidance or experience cannot effectively evaluate contractors' proposed system engineering plans, including proposed requirements engineering activities. Program office engineers generally do not have the experience and training to do the necessary requirements engineering activities described in the guidebook which the Air Force must perform in preparation of system/segment specifications (Type A) before a request for proposal (RFP) is released.

Air Force program offices have increased their use of external systems engineering support contractors (government and private) to produce the engineering products and to support the acquisition process. The external support engineers are also generally without procedural guidance for the definition and analysis of system requirements. Many corporations supporting the system engineering activities of Air Force program offices have years of experience in systems engineering activities of Air Force program offices and are able to translate many of the principles and practices of guidance like the AFSCM 375 series into present regulations, standards, and specification applications. This is one of the reasons the program offices rely upon their support. However, the new Air Force program office engineers are without the experience or guidance for accomplishing requirements engineering, or for managing and monitoring the progress of systems engineering support staffs or the requirements engineering efforts of development contractors.

Review of specifications being prepared by program offices, systems engineering support contractors, and development contractors clearly demonstrates the varying quality of specifications being written. Current

guidelines, specifically MIL-STD-490, MIL-STD-483 (USAF), MIL-STD-499A (USAF), and AFR 800-3, are inadequate and unsuitable for the requirements engineering which is described in the Requirements Engineering Guidebook and which is necessary for the complexities of the systems being acquired. The retreat from the more regulated aspects of the AFSCM 375 series has been too extreme. The Requirements Engineering Guidebook provides a framework in which Air Force program offices, supporting systems engineering contractors, and development contractors can understand the tasks and issues of requirements engineering in the Air Force acquisition life cycle. The guidelines and standards isolate the tasks and issues of defining system requirements within the context of the present Air Force systems acquisition environment. This approach provides the missing procedural guidance which is necessary in understanding the definition and analysis of requirements for Air Force systems. The guidebook is oriented toward filling the need for definitive guidance while providing flexibility in the approach which each application of the guidelines and standards will necessitate.

The definition and analysis of system requirements is usually accomplished in the early phase of the system acquisition life cycle (Conceptual Phase) by Air Force program offices and in a subsequent phase (Validation Phase) under contract to development contractor(s). The Requirements Engineering Guidebook can be used by Air Force program offices to define and analyze the requirements of a system leading to the preparation of Type A system/segment specifications and subsequently by contractors in preparing Type B5 computer program configuration item specifications. Therefore, the implementation of the guidelines and standards must address the guidebook's use by Air Force program offices as well as how to encourage the guidebook's use by system development contractors.

4.4.2 Visibility Factors

The application of the Requirements Engineering Guidebook would require significant work and documentation by Air Force program office engineers. The present lack of direction provides little visibility into the

requirements engineering activities of Air Force program office engineering staffs or their support engineers. The principle program office goal is typically to produce a draft system/segment specification which satisfies program office schedule and milestone objectives. Program office system acquisition life cycle schedules are well recognized as a principle contributor to certain problems during the system acquisition. More often than not the schedule is determined or required by external issues such as obligating funds rather than derived from a thorough evaluation of the actual time necessary for good systems engineering and development.

The Requirements Engineering Guidebook defines the procedures leading to specification preparation and identifies intermediate products such as functional and information hierarchies, and information and control flows. Following the procedures in the guidebook would necessitate a significant amount of work on the part of Air Force program office and supporting engineers. It should be expected that this visibility into the daily accomplishments of the engineers will be resisted by those more comfortable with the anonymity which the current practices allow. It is seldom possible to distinguish any individual's contribution to the end product. The ability to trace the requirements to their sources (source documents) or to understand the thought processes and decisions surrounding the determination and documentation of the requirements is practically impossible at the present time. The current specification preparation process is generally an iterative drafting-review-redrafting process which takes place over many months. The resources (time and personnel) consumed in the process are enormous and wasteful. The medium (specification drafts) is an improper vehicle for interfacing the ideas, concerns, and requirements of a system. Much of the staffs' attention is focused upon the textual descriptions and semantics at the expense of defining and analyzing the requirements of the system. The requirements are tangled in text and lost within paragraphs as well as spread throughout the contents of the specification document. Discrete and well organized requirements are not apparent and are seldom easy to identify in the resulting specification documents. The Requirements Engineering Guidebook, like the concept of AFSCM 375-5, places the emphasis on engineering tasks preceding

the preparation of specifications. The various forms of intermediate documentation required are more suitable to the needs of identifying, recording, and communicating the requirements as they evolve.

Although the guidebook is suited for requirements engineering, it provides, as discussed previously, an uncomfortable degree of visibility into the engineering progress and products leading up to the preparation of specification documents. Resistance to the guidebook should be expected. Similar resistance was experienced with the AFSCM 375 series. The intermediate documentation requirements of AFSCM 375-5 were manually produced and maintained. It is understandable that the burden of the documentation requirements (intermediate documentation such as functional flow block diagrams, requirements allocation sheets, etc.) was realistic; but, it is also understandable that the visibility which AFSCM 375-5 provided was even more discomforting. Implementation of the guidebook must, on the one hand, recognize that intermediate documentation provides the necessary visibility leading to the preparation of good system requirements documents (Type A and B5 specifications) while also recognizing that automated assistance is necessary to reduce the burden in production and maintenance of required intermediate and final system documentation.

4.4.3 Training Factors

The introduction of the guidebook in the Air Force program office environment for use in preparing or managing the preparation of Type A specifications will require training in order to be successful. Although a requirement to apply the guidebook could be directed towards program offices, the lack of training coupled with a certain resistance factor could result in the guidebook being satisfied only minimally, thereby, lessening the benefits which the guidebook can provide. Real implementation of the guidebook in the program office will be successful when the engineering staff is convinced that the application of the guidebook provides a vehicle for quality requirements definition and analysis which is lacking in the present drafting-review-redrafting

specification practices. If the engineering staff is not convinced that the Requirements Engineering Guidebook provides effective guidance for performing requirements engineering tasks, then a passive attitude will develop. The training must encourage a positive attitude and assist engineers in performing their work. This encouragement must, as previously mentioned, be coupled with certain automated techniques which will facilitate the development and maintenance of the necessary requirements definitions and the production of the necessary intermediate and final documentation. As a result, the analyst will be freed to do those tasks which cannot be performed by applying the guidebook or automated assistance.

Training should be available from the staff level within the various product divisions of the Air Force Systems Command (AFSC). Directives to establish and maintain requirements engineering training should originate within the appropriate AFSC levels. Training should be available through the Air Training Command and other Air Force institutions specifically oriented to systems acquisition training. Where training is required from private sources, the staff elements should arrange for specific training sessions, workshops, and seminars. Program office personnel should be able to receive requirements engineering training through course work at remote training institutions as well as through special courses, workshops, and seminars provided at each product division. The coordination of the training should be a responsibility of product division staff elements and oriented to the types of acquisition problems at each product division. Staff element control provides a means for the continual assessment of training needs and monitoring the success of the application of the guidebook.

Product division staff elements would also be concerned with the application and maintenance of automated assistance to support the application of the guidebook. The implementation of the guidelines and standards and the necessary training would require changes in present practices and responsibilities currently performed by staff elements. Current staff elements are primarily staffed to review the products of the

program office. The coordination and establishment of training and other support to the application of the guidebook would require that staff elements take a more direct role in program office requirements engineering activities. This more active role would be necessary to establish the guidebook and its initial applications in the program office environment. In addition the involvement of staff elements would provide the necessary feedbacks to maintain the guidebook, update training programs, and improve the capabilities of automated assistance.

4.4.4 Regulations, Standards, Specification Impacts

The application of the guidebook will require changes in current regulations, standards, and specifications. However, these changes are considered minor. They are essentially refinements and clarifications. The concept of specification development (Type A, Type B, etc.) is a well established procedure for successive refinement of "needs" which leads to a "design to" concept and ultimately to the realities of an "as built" product. The Requirements Engineering Guidebook is supportive of this system acquisition concept. During this study there were no reasons identified to alter this basic iterative concept. The literature search provided no other concepts which dispute the phased approach which has evolved over the past decades in military systems development. The literature search along with the development of the guidebook tasks did clearly point out that the present contents of the MIL-STD-490 and MIL-STD-483 (USAF) for specification practices are essentially adequate forms of system engineering documentation. However, the required formats for specification preparation within these current standards could be improved, especially with consideration to automated documentation and specification generation from computer maintained requirements data bases. Until such changes are actually made in these two military standards, the present format requirements can be utilized as described in the Requirements Engineering Guidebook (Volume III, paragraph 4.13, Prepare Specification Documentation). In order to require the guidebook to be applied in program office Type A specification preparation, some form of directive would be required, either from the product division level through

command or staff elements or from higher levels within AFSC. Another possibility is for the Program Management Directive to specify that requirements engineering be accomplished in accordance with the Requirements Engineering Guidebook prior to the RFP being issued. The program office would then plan for the resources required and provide the time in their schedules to perform requirements engineering. The planning results would then be incorporated into the Program Management Plan (PMP). This directed approach should not be taken unless the staff elements and training and support capabilities are available. The significant advantage of directing the application of the guidebook is that the program office would have to plan and schedule for requirements engineering. The program office would be following consistent guidance which provides visibility into the activities of the engineering staffs, intermediate documentation, and a vehicle for defining and analyzing the requirements prior to the preparation of the actual specifications released with the RFP.

Various Air Force quality assurance requirements and guidelines, such as AFR 74-1 and product division guidelines (e.g., ESD 74-1), would also require changes. Currently quality assurance is a passive activity with respect to the specification preparation within ESD. The current tasking (ESDM 74-1, Task 5, Specification Review) is concerned with reviewing the specification as an end product in the requirements engineering process. The Requirements Engineering Guidebook provides a greater degree of visibility into the evolution of system requirements. The intermediate documentation requirements would provide quality assurance personnel with the ability to evaluate the progress of systems requirements definition and analysis and to ensure that the requirements are clearly stated, and unambiguous. Although the role of quality assurance personnel is not to question the technical merits of engineering requirements, the quality assurance staff is responsible for ensuring that the specification satisfies certain quality requirements. ESDM 74-1, Task 5 should be modified to incorporate quality assurance review of the program office engineering activities while applying the Requirements Engineering Guidebook to ensure that the guidelines and standards are being applied properly and the intermediate products satisfy quality assurance objectives.

Changes are also necessary to the engineering management concepts required in MIL-STD-499A (USAF) and AFR 800-3. MIL-STD-499A describes the fundamental concepts and criteria against which contractors can propose their individual internal procedures as a means of satisfying Air Force engineering requirements. The System Engineering Management Plan (SEMP) prepared by the contractor provides a means for the contractor to satisfy MIL-STD-499A while taking advantage of his internal procedures in support of Air Force programs. MIL-STD-499A does not specifically address requirements engineering. The definition of requirements engineering presented in this study should be incorporated into MIL-STD-499A. Other changes should direct the contractor to address requirements engineering in the SEMP. The SEMP data item description (DID) should also be modified to ensure that requirements engineering is addressed in the SEMP. Until these changes are made, staff elements or program offices should modify the SEMP DID through back up-sheets. If the SEMP is evaluated as a basis of contract award, the requirements engineering approach will become a more significant activity in the contractors engineering practices.

Finally, the system program office acquisition management guidance (AFR 800-3, Engineering for Defense Systems) should be modified in conjunction with MIL-STD-499A. Again the definition of requirements engineering should be incorporated into AFR 800-3. Specific direction for the program office to perform requirements engineering should also be included as a separate acquisition management task. This specific direction would be the basis upon which the PMD directive for requirements engineering could proceed.

4.4.5 Evolutionary Approach

Success with the application of the Requirements Engineering Guidebook may lead to further refinements and eventually to an Air Force military standard. This should come about in an evolutionary manner to allow time for the application of the concepts of the guidebook within various product divisions. As lessons are learned, the guidebook should be modified to ensure that the practices are current with the system acquisition needs and objectives as well as responsive to other issues. The recommended

approach to the application of the Requirements Engineering Guidebook is to provide the guidelines and standards as an example of a desired approach for requirements engineering and encourage its use. The guidebook could be provided to potential contractors for their consideration in preparation of SEMPs either informally or formally (as in the case of modifying the SEMP DID).

Potential contractors could use the Requirements Engineering Guidebook in proposal preparation if the SEMP were a requirement for evaluation during source selection. This approach is not unlike the method taken with the structured programming series (RADC-TR-74-300) application as it was initially applied in ESD requests for proposals. The contractor proposed Computer Program Development Plans (CPDPs), which are similar in intent to the SEMP but address software development issues, were influenced by the RADC-TR-74-300 series, because the series was either referred to directly in the RFP or included or referenced in the notes section of the system specifications. Eventually the need for guidance which tailored the RADC-TR-74-300 series more directly to the needs of Air Force contracting became a necessity. As a result, the RADC Computer Software Development Specification (CP0787796100D) and other software development standards for modern programming practices in Air Force acquisitions are evolving. A similar evolutionary approach is recommended for the Requirements Engineering Guidebook.

4.5 Automated Tool Design Approaches

4.5.1 Introduction

This subsection describes the current capabilities of an existing Air Force requirements engineering tool, CADSAT. The list of automated requirements engineering tool capabilities (Appendix D) has been reduced to a table format (Table 2). Each tool capability has been evaluated against the basic CADSAT (BC) capabilities as well as the Logicon-Extended CADSAT (EC) capabilities. The requirements engineering tool list has been developed in relation to the requirements engineering activity description in the

TABLE 2 Automated Tool Design Approaches

The following pages provide an evaluation of CADSAT against the recommended Automated Requirements Engineering Tool Capabilities resulting from this study. This table consists of two primary columns: CADSAT Evaluation Ratings and Automated Requirements Engineeringj Tool Capabilities. The CADSAT Evaluation Ratings column is further divided into two subcolumns; one for the University of Michigan CADSAT (Basic CADSAT -BC) and the other column for the Logicon-Extended CADSAT (EC). Within each BC and EC column, there are four possible evaluation ratings as denoted at the top of each column: 1, 2, 3, 4. The meaning for each subcolumn is as follows:

- Column 1: If column 1 contains a BC and/or EC notation, the tool capability at the right is a current BC and/or EC feature.

- Column 2: This column contains an effectiveness evaluation as denoted by the legend notation contained herein. These evaluations indicate the effectiveness of the column 1 tool (i.e. BC or EC) relative to the Automated Requirements Engineering Tool Capabilities column at the right.

  blank - adequate
  VL - very low effectiveness
  LE - low effectiveness
  ME - medium effectiveness
  HE - high effectiveness

- Column 3: This denotes that the current tool (BC or EC) would have to be modified or new capabilities added to the current tool features to achieve the desired Requirements Engineering Tool Capability described at the right. The notations are:

  AD - addition to present capabilities
  MD - modification to present capabilities

- Column 4: This column provides a rough estimate of the cost magnitude associated with the recommended addition or modification for the current tool (BC or EC). The notations are:

  L$ - low costs
  M$ - medium costs
  H$ - high costs
  ?$ - varying costs

TABLE 2 Automated Tool Design Approaches (cont'd)

CADSAT Evaluation Ratings 1 2 3 4	1 2 3 4	Automated Requirements Engineering Tool Capabilities (see Appendix D)	
		1.	LANGUAGE
BC MD ?$	EC	1.1	Objects (50 character names)
BC	EC	1.1.1	Functional Requirements (functions)
BC LE MD ?$	EC	1.1.2	Constraint Requirements
BC	EC	1.1.2.1	Performance
BC	EC	1.1.2.2	Physical
BC	EC	1.1.2.3	Operability
BC	EC	1.1.2.4	Design
BC	EC	1.1.2.5	Test (see 1.1.7)
BC	EC	1.1.3	Sources
BC LE MD ?$	EC	1.1.4	Unique Identifiers (see also 2.11)
BC	EC	1.1.5	Tracekeys
BC	EC	1.1.6	System I/O
BC	EC	1.1.6.1	Physical structure
BC	EC	1.1.6.2	Logical structure
BC	EC	1.1.7	Test Points
BC	EC	1.2	Relationships
BC	EC	1.2.1	Hierarchical Relation of Functions
BC	EC	1.2.2	Association: Constraints to Functions
BC	EC	1.2.3	Related-Requirement Relationship
BC	EC	1.2.4	Association: Sources to Requirements
BC	EC	1.2.4.1	Primary Sources
BC	EC	1.2.4.2	Secondary Sources
BC	EC	1.2.5	Association: Tracekeys to Requirements
BC	EC	1.2.6	Information-flow Relationships:
BC	EC	1.2.6.1	"Uses/used by"
BC	EC	1.2.6.2	"Derives/Derived by"
BC	EC	1.2.6.3	"Updates/Updated by"

TABLE 2 Automated Tool Design Approaches (cont'd)

CADSAT Evaluation Ratings								Automated Requirements Engineering Tool Capabilities (see Appendix D)	
1	2	3	4	1	2	3	4		
BC				EC				1.2.6.4	"Provides/provided by"
BC				EC				1.2.6.5	"Receives/received by"
BC				EC				1.2.7	Control-Flow Relationships:
BC				EC				1.2.7.1	"Triggers/triggered by"
BC	LE	MD	?$	EC				1.2.7.2	"Conditional trigger/conditional triggered by
BC				EC				1.2.7.3	"Utilizes/utilized by"
BC				EC				1.2.8	Hierarchical Relation of System I/O
BC				EC				1.3	Requirements Text
BC				EC				2.	ANALYZER (to be used interactively)
BC				EC				2.1	Change Data Base Reports
BC				EC				2.1.1	Input Report (update/no updated)
BC				EC				2.1.1.1	Check syntax of Requirements Definition
BC				EC				2.1.1.2	List Requirements Definition Errors
BC				EC				2.1.1.3	Requirements Definition input check option
BC				EC				2.1.2	Delete Objects or Relationships Report
BC				EC				2.1.2.1	Check legality of deletion
BC				EC				2.1.2.2	Record deletions and errors
BC	LE			EC	HE			2.1.3	Rename Objects Report
BC	LE			EC				2.1.3.1	Check legality of rename
BC	LE			EC				2.1.3.2	Record renames and errors
BC	LE			EC				2.1.4	Change Object Type Report
BC	LE			EC				2.1.4.1	Check legality of change type
BC	LE			EC				2.1.4.2	Record change types and errors
BC				EC				2.2	Object Information Report
				EC				2.3	Source Document Summary Report

TABLE 2 Automated Tool Design Approaches (cont'd)

CADSAT Evaluation Ratings								Automated Requirements Engineering Tool Capabilities (see Appendix D)	
1	2	3	4	1	2	3	4		
BC	LE			EC	HE			2.4	Functional Hierarchical Structure Report
BC	LE			EC	HE			2.4.1	Functional Hierarchical Structure
								2.4.2	Information available at Analyst Option:
BC	LE			EC	HE			2.4.2.1	Sources (primary and secondary)
BC	LE			EC	HE			2.4.2.2	Associated constraints
BC	LE			EC	HE			2.4.2.3	I/O and relationships: use/derive/update
BC	LE			EC	HE			2.4.2.4	Functional control relationships: triggers/conditional triggers/UTILIZES
BC	LE			EC	HE			2.4.2.5	Using Activities and relationships: provides/receives
BC	LE	MD	L$	EC				2.4.2.6	Requirements Text
BC	LE	MD	?$	EC				2.5	I/O Hierarchical Structure Report
BC	LE			EC				2.5.1	Beginning with a superior member
BC	LE			EC				2.5.2	Beginning with a selected member
BC	LE			EC				2.6	I/O-Function Interaction Report
BC	LE	AD	L$	EC				2.7	Information Flow Report
BC	LE			EC	ME	AD	M$	2.8	Control Flow Report
								2.9	Identify Specified Objects Report
BC	LE	MD	?$	EC				2.9.1	Identify Functional Requirements (functions)
BC	LE	MD	?$	EC				2.9.1.1	All Functions
BC	LE	MD	?$	EC				2.9.1.2	All Functions below specified function
BC	LE	MD	?$	EC				2.9.1.3	With/without associated constraints
BC	LE	MD	?$	EC				2.9.1.4	which use/derive/update selected I/O
BC	LE	MD	?$	EC				2.9.1.5	Which do not use/derive/update selected I/O
BC	LE	MD	?$	EC				2.9.1.6	With control-flow relationships: triggers/conditional triggers/UTILIZES
BC	LE	MD	?$	EC				2.9.1.7	Without control-flow relationships
BC	LE	MD	?$	EC				2.9.1.8	Functions at the lowest level in hierarchy
BC	LE	MD	?$	EC				2.9.1.9	(same as above) below selected Function

TABLE 2 Automated Tool Design Approaches (cont'd)

CADSAT Evaluation Ratings 1 2 3 4	1 2 3 4	Automated Requirements Engineering Tool Capabilities (see Appendix D)	
BC LE MD ?$	EC	2.9.1.10	Functions without source references
		2.9.2	Identify Constraints
	EC LE AD L$	2.9.2.1	All constraint requirements
AD L$		2.9.2.2	Constraints without associated Functions
AD L$		2.9.2.3	(same as 2.9.2.1) of a selected type
BC LE	EC	2.9.3	Identify System I/O
BC LE MD L$	EC	2.9.3.1	All I/O or any I/O below a selected level
BC LE MD L$	EC	2.9.3.2	I/O neither used/derived/nor updated
BC LE MD L$	EC	2.9.3.3	I/O between selected levels in I/O structure
BC LE MD L$	EC	2.9.3.4	I/O used/derived/updated by selected functions
BC LE	EC	2.9.4	Identify Source References
		2.9.4.1	All source references
BC VL	EC	2.9.4.2	Without associated Functions or Constraints
BC VL	EC	2.9.4.3	With a specified prefix
BC VL	EC	2.9.4.4	With a specified suffix
BC	EC	2.9.5	Identify Tracekeys
BC VL AD L$	EC	2.9.5.1	All tracekeys
BC VL AD L$	EC	2.9.5.2	Without associated Functions or Constraints
BC VL	EC	2.9.6	Identify Test Points
		2.10	Find Associated Requirements Report
		2.10.1	Requirements which use/derive/update I/O relative to a selected I/O
		2.10.2	Functions which trigger/are triggered by/UTILIZED a selected Function

TABLE 2 Automated Tool Design Approaches (cont'd)

CADSAT Evaluation Ratings								Automated Requirements Engineering Tool Capabilities (see Appendix D)	
1	2	3	4	1	2	3	4		
								2.10.3	Requirements related by Constraints to a Function
								2.10.4	Requirements having references to a selected function
								2.10.5	Requirements associated by Related-Requirement
								2.10.6	Requirements immediately superior in functional hierarchy
				EC	LE	MD	M$	2.11	Unique Requirement Number Derivation
BC	LE			EC	ME	MD	M$	2.12	Requirement Document Reports
				EC				2.13	Requirements Traceability Reports
				EC				2.13.1	Trace between source documents and a requirements data base
				EC				2.13.2	Trace between one requirements data base (unique identifier/requirements name) and another requirements data base (unique identifier/source reference
				EC				2.13.3	Trace between one requirements data base (source reference/associated requirements name) and another requirements data base (unique identifier /source reference)
				EC				2.13.4	Requirements (sorted on source reference/ unique identifier) not traced to second requirements data base
BC	VL	AD	L$	EC				2.14	Test Reports
BC	ME			EC	HE			2.15	Requirements Data Base Status Reports

guidebook (Volume III). The list represents the minimum capabilities necessary to support the guidebook. Basic CADSAT satisfies most of the guidebook's requirements engineering needs, especially the language capabilities. General deficiencies are in the report generation area (Table 2, 2.0 Analyzer). These deficiencies are both human engineering and system engineering problems.

The Logicon-Extended CADSAT was developed in conjunction with the surveillance system requirements engineering activities. Since this work concentrated upon early requirements engineering (Type A and Type B5 specifications), the enhancements to the basic CADSAT satisfy a number of essential early requirements engineering needs. The extensions concentrate on the analyst's needs for improved report generation. Liberties with the language were taken; the language features were employed differently in some cases than originally intended. The basic and the extended CADSAT satisfy most of the needs of the capabilities list and therefore the guidebook. One approach would be to employ the basic CADSAT with certain extended CADSAT features, primarily to increase ease of use and provide additional reports. The second approach would be to add additional capabilities (reports) as well as to achieve the objectives of the first approach. These improvements would build upon the present design of CADSAT including the Logicon extensions. Each of these approaches is described in the following paragraphs.

4.5.2 Partially Extended CADSAT Approach

Table 2 is a condensed list of the automated requirements engineering tool capabilities discussed in 4.2 and Appendix D. Basic CADSAT (BC) is essentially equivalent to the current University of Michigan User Requirements Language/User Requirements Analyzer (URL/URA), Version 3.3. Extensions to the basic CADSAT made to support Logicon's surveillance system requirements engineering activities are referred to as the Logicon Extended CADSAT (EC). Table 2 notations (BC and EC) indicate the present CADSAT capabilities which satisfy the automated requirements engineering tool capabilities list (Appendix D).

Four essential Extended CADSAT capabilities are considered vital for any application of CADSAT to the requirements engineering activities described in the guidebook. First, the Requirements Data Base Status Reports provide essential technical and management visibility into the development of the requirements data base. Second, the Extended CADSAT structure report (Functional Hierarchical Structure Report, Table 2, 2.4) provides the analyst with the option of printing out portions of the functional hierarchy. The basic CADSAT requires the entire structure to be printed. For large structures, as in the case of most Air Force systems, the ability to select portions of the hierarchy is necessary to conserve resources (print time, etc.) as well as provide the analyst with only the information he desires. Third, the extended CADSAT provides the capability to globally rename parts of the names of an object (Table 2, 2.1.3). For example, if there were numerous object names beginning with "XXX-" and the analyst wished to change all of them to "IFF-" the extended CADSAT rename provides this capability. This is extremely useful if erroneous requirements have been entered. The use of this extension greatly reduces the time necessary to correct the error and the problems in the requirements data base which are incurred from entry errors. Finally, the extended CADSAT provides the capability to trace between requirement data bases (Table 2, 2.13). This extended feature is necessary when the tool is employed in requirements traceability between successive specification documents such as MIL-STD-490/483 (USAF), Type A, B and C specifications. With these four enhancements to CADSAT, the analyst is capable of performing the requirements engineering activities of the guidebook. Other refinements in the extended CADSAT facilitate the use of CADSAT by decreasing the time to do analysis activities, such as information retrieval and report generation, and make it easier for the novice to use the tool more effectively. Therefore, the extended CADSAT with the four features described above satisfies the automated requirements tool capabilities list in a basic sense. However, other additions and modifications are considered essential to improve the human engineering and system engineering concepts of the basic CADSAT and extensions as indicated in Table 2.

4.5.3 Additional Extended CADSAT Approach

The following paragraphs describe numerous refinements, modifications, and additions to CADSAT which are considered essential to fully complement the Requirements Engineering Guidebook. Each of these extensions is explained in reference to Table 2.

CADSAT limits the name of an object to 30 characters. In applying the tool to the surveillance system, the need for additional characters was desired in many instances. A 50-character length naming capability (Table 2, 1.1) is desirable. The magnitude of the changes would primarily affect the existing CADSAT reports. Current CADSAT report formats allow a maximum of 30 characters in the output field for the object name. It is estimated that the use of variable length records for the object names would provide the desired 50-character name maximum while the average length name in actual use would fall below the present fixed length 30-character names.

Constraint requirements (Table 2, 1.1.2.-1.1.2.5) can be handled by CADSAT by using the present MEMO, EVENTS, ATTRIBUTES, and HAPPENS objects. The MEMO object is used extensively in the surveillance system application for identifying system performance constraints (Table 2, 1.1.2.1). However, these CADSAT constraint language features are considered to be awkward and do not adequately describe the various characteristics of each constraint requirement. In addition, the presentation of the constraint requirements in various reports should be achieved in conjunction with enhancements to the language feature.

Unique identifiers are currently handled by the basic CADSAT (Table 2, 1.1.4) and are enhanced by the extended CADSAT (Table 2, 2.11). Additional improvements are considered necessary primarily in allowing some optional capabilities on how the unique identifiers are stored and related to the requirements in the requirements data base. Presently these identifiers are only auxiliary numbers. The option to allow the analyst to permit the identifier to be related to specification document paragraph numbering sequence would support automated requirements document generation (Table 2, 2.12).

CADSAT provides limited capability for "conditional triggers" (Table 2, 1.2.7.2 and 2.8). The CADSAT language feature, CONDITION, provides the capability to describe boolean ANDs and ORs. For example, "Condition, Hardware-fault becomes true, triggers hardware-diagnostics and report-hardware-fault, or becomes false, triggers function B." Changes should be accomplished by identifying various ways the language features and reports could be improved. Test examples should be worked out to identify the human engineering and system engineering requirements which satisfy "conditional trigger" requirements.

The CADSAT Contents Report (I/O hierarchical structure report, Table 2, 2.5) should be extended to allow the user the option of getting a report similar to the Logicon-Extended Structure Report. This extension should present source references for each I/O type, and statements as appropriate for each I/O type, such as derived by, updated by, etc. These enhancements would consolidate useful information now contained in separate CADSAT reports. For example, Contents Report plus a Formatted Problem Statement Report on a designated I/O requirement name would be equal to the information which could be presented in this proposed Extended Contents Report. The combined information, like the Logicon-Extended Structure Report, would be more useful in practice than the current numerous, and less-informative basic CADSAT reports.

The University of Michigan PSL/PSA extended picture report (Version 3.3) provides the capability to present information flow (Table 2, 2.7). The extended picture report format is considered to be of limited use both for information flow and control flow (Table 2, 2.7, 2.8). Picture reports do not provide a condensed presentation of flows since numerous picture reports are required. Each report presents limited information. A condensed listing showing the flow and associated functions, flow conditions, source references, etc. should be developed. One possibility is an indented format similar to the structure reports. Picture reports are more useful in presenting the flows to the novice analyst or to others where the need exists for a quick, graphic understanding of the flow, especially top level flows. Condensed listings are more useful to the trained analyst and provide a quick reference format.

The CADSAT analyzer should be modified to aggregate lower level I/O and associate the aggregates to higher level functions. The present analyzer cannot do this and therefore generates numerous errors when attempting to report the information flow at higher levels in the I/O hierarchy. The higher level information flows are necessary when the analyst desires an overview of the information flow in a particular area in the requirements data base.

Three changes should be made in the CADSAT control flow reports (Table 2, 2.8). The report should provide the capability to trace the control flow forward or backward by a specified number of functions from a given function. Another addition would be to have CADSAT control-flow reports show the relationships of lower level functions to other lower level functions when presenting control flows at higher levels in the functional hierarchy. This is similar to the aggregation deficiency of the information flow described previously. The CADSAT upper level control flow report (CADSAT Process Chain) does not have the capability to elevate lower level functions into higher level flows such that the higher level flows are meaningful. The third addition to the control flow is the capability of the analyzer and reports to handle control flow loops.

CADSAT has basic query capabilities which in conjunction with report generation capabilities can identify specific objects in the requirements data base (Table 2, 2.9). However, the query-report process is not satisfactory for the requirements engineering process involving large data bases. The query-report capabilities should be improved and the specific features identified in Table 2 (2.9.1.1-2.9.1.10) should be added at a minimum.

CADSAT reports which present constraint requirements (Table 2, 2.9.2) do not presently exist. The constraint requirements are presently integrated into the function reports such as the functional hierarchical structure report. All constraints can be identified within the limited CADSAT query-report capabilities. Minor additions to CADSAT could be performed to provide constraint reports to a limited degree without significant effort. More extensive constraint reports would necessitate additional resources.

The CADSAT report capabilities for identifying source references (Table 2, 2.9.4) are considered adequate for the basic application of CADSAT. The present capabilities do not permit limited printouts for selected areas in the requirements data base. No reports are available which provide the capabilities identified in Table 2, 2.9.4.1-2.9.4.4. The latter can be derived from extensive use of the CADSAT query-report capabilities and the resulting information can be consolidated manually. Improvements would increase the ease of use when dealing with source documentation and accomplishing consistency and completeness analysis. The identification of tracekeys (Table 2, 2.9.5) is also considered adequate, but could be improved in the same ways as the source references described above. Improvements to each of these would facilitate the use of CADSAT and provide for improved efficiencies in the requirements engineering process.

Test point identification is a capability which is accomplished by the CADSAT language as described previously (Table 2, 1.1.2.5, and 1.1.7). CADSAT reports provide a very limited capability to present this information (Table 2, 2.9.6 and 2.14). The capability is limited to query-reporting and manually composing the results for a specific test report purpose. At a minimum, test points should be associated with information and control flows. Additional reports specifically oriented toward presenting test information in various formats for analysts and management needs should be defined and developed to satisfy specific test report objectives.

As previously mentioned CADSAT has limited query-report capabilities. The ability to readily retrieve information to identify associated requirements (Table 2, 2.10) requires significant effort on the analyst's part. Extensions to CADSAT in the query-report area should provide the capabilities to readily retrieve the information identified in Table 2 (2.10.1-2.10.6). This capability is essential in analyzing impacts during or subsequent to the requirements data base development.

CADSAT has a limited capability to produce requirements specification documents (Table 2, 2.12). Extended CADSAT has improved capabilities, but none of the extensions provide the capability to automatically generate specification documents directly from the requirements data base in accordance with specific format requirements such as MIL-STD-490/483 (USAF). The relationships between the requirements engineering intermediate documentation (structures, flows, etc.) and paragraphs of MIL-STD-490/483 specification documents (Types A and B5) are presented in Volume III. Enhancements to CADSAT to automatically achieve the translation of a target systems requirements contained in a requirements data base to a MIL-STD-490/483 format are considered to be of minimum technical difficulty. The primary effort would be in satisfying many small details to achieve the required military standard formats. A moderate enhancement with the essential capabilities to generate skeleton MIL-STD-490/483 specification documents directly from the requirements data base could be achieved in a matter of weeks. Additional full capabilities would involve several months of effort. Future enhancement to CADSAT to generate other requirements documents which satisfy other format requirements, such as DoD 7935.1-S can be accomplished based upon the design concepts of CADSAT extensions to satisfy MIL-STD-490/483 requirements.

SECTION 5 RESULTS AND RECOMMENDATIONS

5.1 Introduction

This section presents results and recommendations concerning the Requirements Engineering Guidebook, the role of automated tools in support of the guidebook, enhancements to CADSAT, extended applications of CADSAT and the use of the requirements data base, the approach to the application of the guidebook, and the means of facilitating the application of the guidebook and automated tools within various acquisition environments.

5.2 Results

5.2.1 The Requirements Engineering Guidebook

The application of the Requirements Engineering Guidebook should come about in an evolutionary manner to allow time for the application of the concepts within various acquisition environments. As lessons are learned, the guidebook should be modified to ensure that the practices are current with the Air Force system acquisition needs and objectives. The guidebook should become the basis for defining automated tool capabilities. The automated tool capabilities list and subsequent specifications, such as DoD Standard 7935.1-S functional description, system specification, and program specification, should also be the baseline for the functional definition and design of tools including the upgraded CADSAT. This method of baseline management should be achieved for the requirement engineering guidelines and standards, and automated tools as it would be for any other Air Force system. The resulting configuration and documentation control would ensure that a system like CADSAT can be operationally deployed in the various using agencies and contractor acquisition environments.

5.2.2 Requirements Engineering Tools

The emphasis of the Requirements Standards Study has been on describing the goal of requirements engineering, the characteristics of quality

requirements, and the principles and practices of requirements engineering. Automated assistance (CADSAT) has been presented as a tool for defining and analyzing system requirements. Existing automated tools lack many of the fundamentals of requirements engineering from an Air Force systems engineering viewpoint. The origins of these tools within academic and R&D environments and the lack of pragmatic applications to complex military systems development are evident in the performance and design of the initial tools. Logicon CADSAT extensions have resolved many of the needs for an early requirements engineering tool and have been effective in Air Force systems acquisition requirements engineering activities. Additional needs have been identified during this study. A requirements engineering tool like CADSAT is most effective in the early acquisition phases of system development in conjunction with the guidelines and standards for requirements engineering such as the guidebook developed during this study.

5.2.3 Language Proliferation

The current trend in development of automated tools is toward more specialized requirements engineering features for each acquisition environment. Many user desire unique language features and report capabilities which reflect the actual uniquenesses of his acquisition environment. This trend will most likely continue unless a policy decision places limits on the proliferation of unique tools. One conclusion of the RSS is that the proliferation of unique tools is unnecessary. The approach that should be taken is to define the guidelines and standards for requirements engineering throughout the Air Force (as well as DoD). Next the automated aids such as those described in this study can be implemented and applied in response to the guidebook. Finally, specific requirements in various acquisition environments can best be addressed by development of specific requirements engineering methodologies on the application of the guidelines and standards and tool to a specific environment. For example, the guidebook requires that functions be described and organized into a hierarchical structure. A requirements engineering tool like CADSAT provides the capability to define the functions and structure the functions hierarchically. A specific methodology for an electronic system acquisition

might require top-level function names such as surveillance, tracking identification, interceptor control, and communications. A methodology for space system acquisition environment might show how to apply the tool and define system functional requirements with top-level function names such as flight mission, launch, and status monitoring. The proliferation of the language could be taken to the extreme with language objects such as trackers, missiles, sensors, or the like.

Therefore, the guidebook and automated tool can serve any acquisition environment and the application of the guidebook and use of automated tools is best addressed through specific guidance - a requirements engineering methodology. The guidebook and list of requirements engineering tool capabilities, such as those described in this study, should be the initial requirements documents for an automated tool. The tool is the response to the implementation of the guidelines and standards. The methodology is the user's guide which addresses the specific application of the guidebook and tool to each acquisition environment.

5.2.4 Language Simplification

The CADSAT language features generally satisfy the requirements of the Requirements Engineering Guidebook. Many of the present language objects can be eliminated, consolidated with other objects, or renamed. For example, the PROCESS should be renamed FUNCTION. The number of objects for identifying system I/O (SETS, ENTITIES, INPUTS, OUTPUTS, GROUPS, ELEMENTS) can be reduced to fewer objects such as SETS (where a set can be any grouping of other I/O objects), INPUTS and OUTPUTS (external and internal I/O) and ELEMENTS (the last identifiable I/O component in the hierarchy). The INTERFACE object is merely an external system function and is a confusing object name. One possibility would be to rename INTERFACE to something more representative of an external function. These example changes are intended to give only an idea of the possible simplification of the CADSAT language which would conform to the Requirements Engineering Guidebook. The actual concepts and design should be described in a specification for a revised CADSAT language such as the Functional

Description (FD) and subsequently the system specification (SS) of the DoD Standard 7935.1-S [16]. This streamlining of the language would provide clarity in understanding the capabilities of the tool language and greatly simplify the training and application of the tool in support of the standard.

5.2.5 Analyzer Refinements

The CADSAT analyzer, including the current Logicon extensions, satisfies most of the requirements engineering activities described in the guidebook The improvements to the analyzer described in 4.5.3 are considered essential to complement the Requirements Engineering Guidebook. Each of these CADSAT analyzer requirements, modifications, and additions should be expanded in the FD and SS along with language features described above. CADSAT reports which do not support the guidebook should be eliminated from the upgraded CADSAT.

5.2.6 Existing Tool Performance and Design

Concurrent with streamlining the CADSAT language and analyzer features to satisfy the standard, various other CADSAT deficiencies should be evaluated and improvements should be presented in the FD and SS. These improvements include requirements data base management, multiple requirements data bases, software segmenting, resource requirements (processor time, memory requirements) and the like. The revised CADSAT configuration and design should improve the effectiveness of the tool's application and ease of use. The FD and SS should address both human engineering and system engineering issues (CADSAT as an automated system) and form the functional-performance baseline upon which CADSAT complies with the guidebook. This baseline should provide the basis for configuration control of the tool, its operational use, and further enhancements.

5.2.7 Simulation

Simulation was addressed in the Requirements Engineering Guidebook and

associated with appropriate requirements engineering activities. However, it has been omitted from the list of tool capabilities. Simulation of system requirements requires a basic set of well-defined system requirements. Simulation requires an extensive allocation of resources: computer time, memory, software and analysts' time. If the simulation is driven from a requirements data base which contains a definition for a functional design, the simulation techniques might demand a more extensive requirements language facility since the system definition may have to include more specific details. As the number of language objects and corresponding requirements data base entries increase, the complexity of the tool's use increases. Although simulation is an essential tool in many instances for identifying the characteristics of the performance requirements, it is only another means of analyzing the requirements of the system. Ultimately the results of the simulation analysis must be incorporated into the requirements data base constraints or refinements to existing constraints. The incorporation of simulation capabilities directly into the requirements engineering tool might encourage the analyst to produce simulations which are in themselves perceived to be the system requirements. Safeguards against this tendency would have to be defined and incorporated into the tool and methodology for its application. Existing requirements engineering tool simulation capabilities are not adequately addressed in the available documentation reviewed during this study. The capabilities and products of the simulation are not described in terms of the specific benefits derived by the simulation of the system from the requirements data base.

There is no question that simulation is a useful analytical technique in specific applications for determining the constraint requirements in a system definition. Numerous simulation languages have been developed and successfully applied to systems development over the previous decades. The requirements engineering guidebook supports this concept. However, because of the previously described reasons, it is not considered appropriate to define simulation as a basic tool requirement which should be incorporated directly into a requirements engineering tool capabilities list at this time. Existing simulation capabilities which use the requirement data base

are too experimental and investigatory at this time to be recommended as essential capabilities for requirements engineering. Additional study concentrating upon the role of simulation and its capabilities, limitations, products, benefits, etc., is warranted and should be accomplished. The results of such a study should be reflected in the requirements engineering guidebook, and the corresponding list of tool capabilities, developed into FD and SS specifications and potentially developed and incorporated into CADSAT.

5.2.8 Query-Report Capabilities

The list of requirements engineering tool capabilities developed during this study does not include an extensive definition of query-report features. The list does contain basic query-report capabilities which are considered essential for supporting the guidebook. A more extensive list of query-report features would provide the experienced analyst with greater capabilities to do extensive requirements analysis beyond the essential capabilities provided by the current CADSAT analyzer fixed report and limited query-report features. In addition, the use of the requirements data base for management information system (MIS) applications (configuration management, traceability analysis, status monitoring, etc.) would be benefited by additional capabilities beyond the essential query-report features. Many of the basic program office MIS needs are directly related to the requirements definition and analysis activities as reported in a previous Logicon study [19]. The utility of additional query-report capabilities should be studied in relation to requirements engineering needs as well as the MIS needs. The query-report capabilities, like the previously described simulation capabilities, should be reflected in the requirements engineering guidebook and the corresponding list of tool capabilities, developed into a FD and SS specification, and potentially developed and incorporated into CADSAT.

5.2.9 Automated Specification Document Generation

The automated generation of specification documents from the requirements data base is a limited Basic CADSAT capability. The Basic CADSAT process depends upon an extensive degree of external analysts' discipline.

For instance, the analysts must ensure that all the desired information is in the requirements data base, define the actual outline of the target document (documentation schema) and prepare and coordinate textual descriptions about the requirements (documentation source). As described in the user's manual the automated documentation system capabilities requires consistency and coordination between the documentation schema, the documentation source, and the requirements data base in order to automatically generate a good document ([14], p. 467). The full potential of the requirements data base is not fully integrated into the generation of the requirements documents.

The Logicon Extended CADSAT, unlike the Basic CADSAT, is able to edit the requirements text as part of the requirements definition and analysis activities. The text is easily edited and reformatted directly from the analyzer. One form of specification documentation can already be directly generated from the requirements data base but is limited to a format which satisfies the ESD surveillance system requirements engineering needs. This extension could be further enhanced to generate specific MIL-STD-490/ 483 documentation by incorporation of different format requirements (schemas) and elimination of certain output fields.

The automated generation of requirements documents should proceed upon the requirements data base as defined and analyzed along the procedures in the Requirements Engineering Guidebook. This extension would encourage the application of the Requirements Engineering Guidebook since CADSAT requires the same forms of definition and analysis leading to specification preparation as are described in the guidebook, i.e., functional analysis, hierarchical structures, control and information flows, etc. The documentation schemas of MIL-STD-490/483 (USAF) should exist within CADSAT. The schemas should be automatically linked to the requirements data base system structures. For instance, the System/Segment Specification functional areas (MIL-STD-490, Type A, paragraphs 10.3.1 and 10.3.7) are identifiable in the upper levels of the functional hierarchy of the requirements data base if developed in accordance with the Requirements Engineering Guidebook. The structure of the hierarchy is directly translatable into the paragraph

numbering and identification for paragraphs 10.3.1 and 10.3.7 and their subparagraphs. The associated constraints, I/O and I/O hierarchies, control and information flows, text descriptions and other requirements can also be directly translated into the appropriate paragraphs of MIL-STD-490/483 as described in the Requirements Engineering Guidebook (Volume III: Tables 2 and 3).

In addition, the design of CADSAT to accomplish MIL-STD-490/483 documentation will quite naturally lend itself to the generation of other forms of documentation such as DoD Standard 7935.1-S and other DoD agency specification requirements with the development of appropriate schemas within CADSAT. The user could then identify which form of documentation he desired and receive an up-to-date version of the documentation desired.

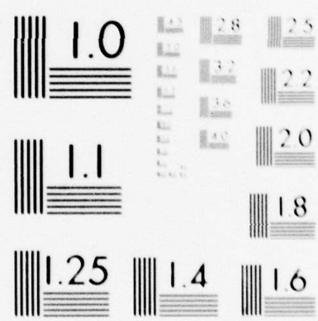
5.2.10 Existing Practices, Regulations, Standards and Specifications

Requirements engineering begins with the first statement of a requirement. This occurs before the development of early *requirements documents*. By the time the program office is formed and the requirements are further analyzed and refined, the rationale for the requirements has been obscured. The analyst in many instances is unable to distinguish a requirement from a desirable feature. Early requirements documents should also be required to identify the source of the requirement and the rationale for the requirement. This information can in turn provide the necessary background information to understand and expand the initial requirements leading to the preparation of a specification document.

Requirements engineering must be a required Program Office engineering activity which is described in AFR 800-3 and MIL-STD-499A (USAF) as previously described (4.4.4). The requirements engineering guidebook or similar guidance should be the basis upon which government and contractor requirements engineering proceeds. Requirements engineering should be an established engineering activity within the program office. The program office quality assurance role should be incorporated into the program office requirements engineering activities. Changes to the quality

AD-A077 848 LOGICON INC LEXINGTON MA F/G 5/1
REQUIREMENTS STANDARDS STUDY. VOLUME II. TECHNICAL REPORT.(U)
OCT 79 D G SMITH, P B MERRITHEW F30602-77-C-0207
UNCLASSIFIED DSJ-R79007-VOL-2 RADC-TR-79-240-VOL-2 NL

2 OF 3
AD
A077848


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

assurance quidelines, such as ESD 74-1, should be accomplished as previously described (4.4.4). Policy for requirements engineering in the program office, similar to ESD 74-1, must also be developed and applied. In addition, training is necessary to support the guidebook and the application of automated tools, if requirements engineering in the program office is to be successful (4.4.3).

5.2.11 Requirements Engineering Methodology

As previously described (5.2.3) a requirements engineering methodology can facilitate the introduction of the Requirements Engineering Guidebook and automated tools in various acquisition environments and provide effective use of the guidebook and an automated tool such as CADSAT. Therefore, the requirements engineering methodology interfaces a general requirements engineering approach such as the Requirements Engineering Guidebook, requirements engineering tools, and the unique needs of each acquisition environment employing the guidebook and requirements engineering tools. The documentation of various requirements engineering methodologies for the ESD environment can proceed at this time from experience gained from the present use of CADSAT in the ESD program office environment. Any extensions to CADSAT should be incorporated into documentation of a requirements engineering methodology.

Similar requirements engineering methodologies should be developed as the guidebook and tools are applied to other acquisition environments such as the missile system, space system, or avionics acquisition environments. In addition to 800-series procurements, requirements engineering methodologies should address the general automated data processing (ADP) procurements within various Air Force acquisition environments.

The various requirement engineering methodologies must provide concise guidance and techniques for requirements engineering within the specific acquisition environment. The requirements engineering methodology must explain the guidebook and tool through succinct text and example applications which provide the user with a direct and simplified

understanding relative to his unique acquisition environment. This approach provides configuration control of automated requirements engineering tools used throughout the Air Force.

5.2.12 Management Information System Capabilities

As previously described, the requirements data base can be useful for management information system (MIS) requirements. The requirements engineering tool serves the analyst during the definition of the requirements by providing a means of storing, retrieving, documenting, and analyzing the requirements as the engineering activities proceed. As the requirements are defined the requirements data base can also be used for MIS purposes. The potential benefits are realized when the requirements data base is the source for such MIS analysis and reporting activities as configuration management, traceability analysis, engineering change proposal evaluation, documentation maintenance, status monitoring, and the like. Extended CADSAT incorporates some of the MIS features required in program office system acquisitions. Additional capabilities can be achieved by a more extensive query-report capability as described previously.

The MIS study [19] describes four areas of program office MIS needs: cost/ budgeting, scheduling, ECP evaluation and control, and plans and contract preparation/control. Engineering Change Proposal (ECP) evaluation and control needs concentrate upon determining the impacts of ECPs on system requirements and the determination of the status of all ECPs in the program office (ECP tracking). The ability of using the requirements data base for change impact analysis and tracking such as described in the Requirements Engineering Guidebook under traceability analysis is a direct application of the requirements data base to program office requirements engineering and MIS needs. Program office planning and contract preparation/control needs to concentrate upon (1) developing a standardized approach for defining the functional and component breakouts of systems, (2) developing a means of assessing the impacts of requirements changes to

established plans and documents (Specifications, etc.), (3) developing a means of identifying the inconsistencies and incompleteness of the system requirements, and (4) the ability to produce plans and contract documents in a timely manner (clerical functions). Each of these MIS needs can be directly supported by the application of the Requirements Engineering Guidebook and automated tools such as CADSAT with the present Logicon extensions and other extended MIS capabilities.

Additional study to describe the relation between the requirements data base and program office MIS needs is recommended. The MIS study [19] and the results of the Requirements Standards Study should be reviewed and the benefits of the use of the requirements data base for MIS purposes should be specifically addressed. The potential for the government or contractor requirements data bases becoming accessible to support program office MIS reporting needs could provide the means of obtaining accurate and current information on the progress of systems acquisition.

5.3 Recommendations

5.3.1 The Requirements Engineering Guidebook

The Requirements Engineering Guidebook (Volume III) has been developed from an analysis of past and present DoD and Air Force system engineering practices. It incorporates established requirements engineering techniques and approaches of many leading defense contracting firms. The Requirements Engineering Guidebook provides the necessary guidance for requirements engineering which is not described in current Air Force regulations, standards, or specifications.

The guidebook provides a general road map for performing system requirements definition and analysis. It begins with definition of the initial user requirements and continues through the complete definition and analysis of the system prior to its development. The guidebook allows for a flexible approach in its application while providing the necessary guidance for government and contractor system analysts to plan and perform requirements engineering activities.

It is recommended that the guidebook be applied to selected programs to allow for clarification and improvement of its contents and presentation. The application of the guidebook as a general guide may lead to a more formalized approach such as direct contract applications or a formal military standard.

The relationship of the Requirements Engineering Guidebook to later phases of the Air Force system life cycle (such as in the full-scale development phase) should be studied and presented as an extension to the Requirements Standards Study. This study would identify the interface between the Requirements Engineering Guidebook, automated requirements engineering tools, and requirements data bases during full scale development, production and deployment, and operations. Such a study would address computer program development incorporating the principles of top-down design and implementation (as an extension to the Requirements Engineering Guidebook), modern programming practices, tools and methodologies, documentation preparation (Computer Program Configuration Item Specifications Type B5), users' manuals, position manuals, technical manuals) and support configuration and data management into the operations phase. The continuation of system test planning and subsystem and system integration testing would also be an integral part of the extension to the Requirements Engineering Guidebook.

5.3.2 CADSAT Enhancements

CADSAT has been found to be an effective tool for accomplishing the requirements engineering activities described in the Requirements Engineering Guidebook. Certain modifications, additions, and improvements to CADSAT have been identified during this study. These enhancements are oriented to improving the human engineering and system engineering of CADSAT as a system which supports the requirements engineering process. The recommended improvements include simplifying the language, streamlining the analyzer to eliminate unnecessary reports, improving existing report capabilities, and increasing the overall performance and design of CADSAT. These enhancements would increase the effectiveness

in support of the application of the Requirements Engineering Guidebook and would improve the CADSAT's efficiency. Extensive use of CADSAT in the Air Force acquisition environment will require continuing enhancements to satisfy additional needs as described below.

5.3.3 Extended CADSAT Capabilities

Four promising uses of a requirements engineering tool like the Logicon-Extended CADSAT are (1) automated specification generation from the requirements data base, (2) management information system applications, (3) additional query-reporting capabilities, and (4) simulation capabilities. The first three are current CADSAT capabilities to a limited extent. Simulation using a requirements engineering tool is considered too experimental to recommend as an essential capability for a requirements engineering tool at this time. Improvements in automated specification generation from a requirements data base is considered the most beneficial enhancement to CADSAT. Extended query-reporting capabilities and management information system features would be the next most beneficial extension beyond the essential capabilities identified to support the guidebook. Finally, simulation should continue to be investigated and experimental approaches encouraged. A thorough analysis of the benefits of simulation in the Air Force acquisition environment, the identification of requirements engineering simulation capabilities, and the development of the specific capabilities into current requirement engineering tools such as CADSAT is recommended as a principle area of research at this time.

5.3.4 Evolutionary Approach

The application of the Requirements Engineering Guidebook and the recommended changes to existing practices, regulations, standards, and specifications must proceed in a careful and selective manner. The guidebook has been developed to satisfy the requirements for a comprehensive approach to defining and analyzing the requirements of a system. The presentation allows for a variable approach to the actual application of the guidelines and standards to satisfy each application

and acquisition environment. However, the requirements for quality requirements definition as addressed in the guidebook are extensive and demanding upon the system analyst. The incompatibility of the guidebook with some current acquisition practices will require changes to existing regulations, standards, and specifications over a period of time to allow the application of the guidebook to evolve. In addition, the resistance factor within government and contractor engineering groups can be expected. Essential to the promotion of the guidebook is adequate training for the engineers who must apply the guidebook to their programs. The key element of this training will be to present the guidebook as guidlines and standards for improved requirements engineering. The benefits of the guidebook must be understood by the analyst who will apply the principles to his work. The success of the documentation and analysis requirements of the guidebook will depend upon the availability of requirements engineering tools like CADSAT. The intermediate documentation and analysis needs which are essential to the requirement engineering process are not considered to be easily accomplished without automated assistance. The availability of automated tools and associated training is essential to the success of the guidebook. In addition, the specific issues of the acquisition environment, the application of the guidebook, and the use of automated tools must be addressed in specific methodologies for each acquisition environment as described below.

5.3.5 Requirements Engineering Methodology

The Requirements Engineering Guidebook presented in this report provides the procedural framework for the definition and analysis of system requirements for any Air Force systems development. The associated list of automated tool capabilities was developed to complement the guidebook and to facilitate the definition, analysis, and documentation of the system requirements. Specific approaches for the application of the guidebook within various acquisition environments, such as ESD, and the integration of automated tool capabilities in support of the requirements engineering tasks described in the guidebook can be facilitated by specific guidance - a requirements engineering methodology. The methodology provides the means

of adapting the procedures and tools to specific acquisition environments and facilitates the introduction of the guidebook and automated tools. The development of a requirements engineering methodology for the ESD acquisition environment based upon the Requirements Engineering Guidebook and automated assistance from CADSAT can proceed as an extension to this study. Additional guidelines for other acquisition environments can proceed based on the ESD methodology.

APPENDIX A REFERENCES

[1] AFSCM 375-5, Systems Engineering Management Procedures, 10 March 1966, rescinded.

[2] AFSCM 375-1 and revisions 375-1A, Configuration Management During Definition and Acquisition Phases, 1 June 1964, rescinded.

[3] "COMSYNCSAT System Study," Systems Engineering Management Workshop, Vol. I, General Electric, The Space Division and the Re-entry and Environmental Systems Division, Philadelphia, PA, c. 15 July 1969, Section 1.1.0, p. 35.

[4] MIL-STD-490, Specification Practices, 30 October 1968.

[5] MIL-STD-483 (USAF), Configuration Management Practices of Systems, Equipment, Munitions, and Computer Programs, 12 April 1971.

[6] AFR 800-3, Engineering for Defense Systems, 30 August 1973.

[7] MIL-STD-499A (USAF), Engineering Management, 1 May 1974.

[8] L.A. Johnson, P.B. Merrithew, and D.G. Smith, "Automated Analysis of System Specifications," The Second U.S. Army Software Symposium, Logicon, Inc., Williamsburg, VA, 26-27 October 1978, pp. 235-258.

[9] DoDD 5000.1, Acquisition of Major Defense Systems, 1 January 1977.

[10] DoDD 5000.2, The Decision Coordinating Paper (DCP) and the Defense Systems Acquisition Review Council (DSARC), 1 January 1977.

[11] DoDD 5000.29, Management of Computer Resources in Major Defense Systems, 26 April 1976.

[12] AFR 800-14, Vol. II, Acquisition and Support Procedures for Computer Resources in Systems, 26 September 1975.

[13] User Requirements Language (URL) User's Manual, Parts I and II, H6180/Multics/Version 3.3, July 1978.

[14] User Requirements Analyzer (URA) User's Manual, H6180/Multics/Version 3.3, July 1978.

[15] D. Teichroew and E.A. Hershey III, "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems," IEEE Trans. Software Eng., Vol. SE-3, pp. 41-48, January 1977.

[16] Standard 7935.1-S, Automated Data Systems Documentation Standards, Office of the Assistant Secretary of Defense, 13 September 1977.

[17] HIPO - A Design Aid and Documentation Technique," IBM Manual, 1974.

[18] D.T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas," IEEE Trans Software Engineering, Vol. SE-3, No. 1, Jan 77, p.18.

[19] Johnson, L.A., Coker, F.T., and Smith, D.G., Management Information System Requirements for ESD Program Offices, Logicon Report No. DSJ-R77024, Contract No. F19628-77-C-0178, USAF, Electronics Systems Division (ESD/TOI), Hanscom AFB, MA, 31 March 1978, p. 164.

APPENDIX B - GLOSSARY

This appendix consists of definitions of the major terms used throughout this document and concludes with a list of acronyms and abbreviations. The definitions are drawn from a variety of sources which are identified at the conclusion of the definition section.

DEFINITIONS

Acquisition Life Cycle - The five phases of system and related item acquisition (Conceptual, Validation, Full-Scale Development, Production and Deployment) with three key decision points (Program, Ratification, and Production Decisions) between each of the first four phases. A program may skip a phase, have program elements in any or all other phases, or have multiple decision points per phase. (AFR 800-2) [1] (See also System/Acquisition Life Cycle). These phases are being redefined [12], [13].

And - Activities preceding the AND must be accomplished before the flow may continue.

Authenticate - The act of signifying (by the approval signature of a responsible person of the procuring activity) that the Government is in agreement with the requirements contained in the specification. Authentication by the procuring activity normally will be accomplished on that issue of the specification which is to be the contractual requirement for the baseline which that particular specification defines (MIL-STD-483 (USAF) paragraph 3.4.9). [2]

Availability - The degree to which the system shall be in an operable and committable state at the start of the mission(s) is called for at an unknown (random) point in time [3]. Reliability and Maintainability are interrelated. The formula used to express this relationship is:

$$A = \frac{MTBF}{MTBF\ MTTR}$$

where

A = Availability
MTBF = Mean Time Between Failure
MTTR = Mean Time to Repair

A figure of merit such as Availability is much more meaningful when applied to systems that operate continuously rather than the use of MTBF. [1] (See also Reliability and Maintainability)

Base Line - A configuration identification document or a set of such documents formally designated and fixed at a specific time during a CI's life cycle. Base lines, plus approved changes from those base lines, constitute the current configuration identification. For configuration management there are three base lines, as follows:

a. Functional Base line. The initial approved functional configuration identification.

b. Allocated Base line. The initial approved allocated configuration identification.

c. Product Base line. The initial approved or conditionally approved product configuration identification. (DOD Directive. 5010.19).[4]

Civil Engineering - This term refers to the Air Force civil engineering functions as they relate to the design, construction maintenance, and operation of facilities necessary to support the acquisition and operation of a system or a major modification program. The impact of the various technical functions on Air Force civil engineering functions must be considered throughout the process of developing and acquiring a supportable and cost-effective system. Civil engineering requirements are derived as a part of the systems engineering process (see AFM 86-1). (See also Engineering Management). [6]

Computer Program - The computer program as it pertains to configuration management is a configuration item defined as a deck of punched cards, magnetic or paper tapes, or other physical medium containing a sequence of instructions and data in a form suitable for insertion into a computer. Computer programs used for administrative purposes and those not associated with system/equipment managed by AFR 65-3 are controlled by AFR 300-2. (See definition under Software). [5]

Computer Program Component (CPC) - A CPC is a functionally or logically distinct part of a computer program configuration item (CPCI) distinguished for purposes of convenience in designing and specifying a complete CPCI as an assembly of subordinate elements. [5], [7]

Computer Program Configuration Item (CPCI) - The computer program as it pertains to configuration management is a configuration item. A CPCI is defined as a deck of punched cards, magnetic or paper tapes, or other physical medium containing a sequence of instructions and data in a form suitable for insertion into a computer. (See also Computer Program) [8]

Computer Program Development Plan (CPDP) - The CPDP is the plan which identifies the actions required to develop and deliver computer program configuration items and necessary support resources. It is prepared by the implementing command or,if the development effort is contracted, the plan may be prepared by the contractor and approved by the implementing command. (AFR 800-14, Vol II) [9]

Computer Program Development Specification - Also called Computer Program Configuration Item Specification, MIL-STD-483 (USAF), see Type B5.

Computer Program Life Cycle - The sequence of activities grouped into phases that characterize the typical process of software production and use. The phases are

Analysis Phase
Design Phase
Coding and Checkout Phase
Test and Integration Phase
Installation Phase
Operation and Support Phase

A particular computer program will undergo these phases at least once during the system acquisition life cycle, however, this may occur entirely in one phase of the system acquisition life cycle (e.g., a mission simulation computer program in the conceptual phase) or over several system acquisition phases (e.g., a mission application program developed over the validation, full-scale development and production phases). See AFR 800-14 Volume II, Section 2-8, for further discussion of the computer program life cycle in the system acquisition life cycle. [8]

Concept of Operations. A verbal or written statement, in broad outline, of a commander's assumptions or intent in regard to an operation or series of operations. The concept of operations frequently is embodied in campaign plans and operation plans, in the latter case particularly when the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give the overall picture of the operation. It is included primarily for additional clarity of purpose and is frequently referred to as commander's concept. (Source: JCS Pub. 1) [13].

Conceptual Phase - The initial period when the technical, military, and economic bases for acquisition programs are established through comprehensive studies and experimental hardware development and evaluation. The outputs are alternative concepts and their characteristics (estimated operational, schedule, procurement, costs, and support parameters) which serve as inputs to the Decision Coordinating Paper (DCP) on major systems, Program Memoranda (PM) on smaller systems/equipment, and to HQ USAF decision documents (Program Management Directives) for programs that do not require OSD decisions. (AFR 800-2) [1] (see also Acquisition Life Cycle

Configuration - The functional and/or physical characteristics of hardware/software as set forth in technical documentation and achieved in a product. (DOD Directive 5010.19) [4]

Configuration Control - The systematic evaluation, coordination, approval or disapproval, and implementation of all approved changes in the configuration of a CI after formal establishment of its configuration identification. (DOD Directive 5010.19) [4]

Configuration Item (CI) - An aggregation of hardware/computer programs of any of its discrete portions, which satisfies an end-use function and is designated by the Government for configuration management. CIs may vary

widely in complexity, size and type, from an aircraft, electronic or ship system to a test meter or round of ammunition. During development and manufacture of the initial (prototype) production configuration, CIs are those specification items whose functions and performance parameters must be defined (specified) and controlled to achieve the overall end-use function and performance. Any item required for logistic support and designated for separate procurement is a configuration item. (AFR 65-3) [1] The third level in the functional hierarchical structure. (See also System Segment, Functional Area, and CPCI)

Configuration Management - A discipline applying technical and administrative direction and surveillance to (1) identify and document the functional and physical characteristics of a configuration item, (2) control changes to those characteristics, and (3) record and report change processing and implementation status. (DOD Directive 5010.19, AFR 65-3, AFR 800-3) [4],[6] (See also Engineering Management)

Constraints - Performance Requirements, Physical Requirements, Operability, Test Requirements, and Design Requirements.

Contractor - An individual, partnership, company, corporation, or association having a contract with the procuring activity for the design, development, design and manufacture, maintenance, modification or supply of items under the terms of a contract. A government activity performing any or all of the above actions is considered to be a contractor for configuration management purposes. [4]

Control Flow (also called Functional Flow) - The description of the logical flow in which the system functions are accomplished in order to control the system functions and satisfy the operational requirements. The control flow indicates only the relationship between system functions and does not imply any lapse in time or intermediate activity. Conditions which determine the flow directions are described using the control-flow relationships: SERIES, AND, OR, and UTILIZES.

Decision Coordinating Paper (DCP).- The principle document to record essential system program information for use in support of the Secretary of Defense/Secretary of the Air Force decision making process. A DCP intended for final approval by the Secretary of the Air Force is called an Air Force Decision Coordinating Paper (AFDCP). (Ref: AFR800-2) [13]

Deficiency - Operational need minus existing and planned capability. The degree of inability to successfully accomplish one or more mission tasks or functions required to achieve mission or mission area objectives. Deficiencies might arise from changing mission objectives, opposing threat systems, changes in the environment, obsolesence, or depreciation in current military assets. [13]

Dependability - Dependability addresses the issues of system survivability, vulnerability (S/V) and external electromagnetic interference. Survivability is the ability of the system to achieve its mission under the conditions of a man-made hostile environment. In addition the system may

be required to operate under the conditions of interference from external electromagnetic sources (Electromagnetic compatibility) as well as operate under threat of possible electronic countermeasures (ECM) such as spoofing and jamming.

Deployment Phase - The period beginning with the user's acceptance of the first operational unit and extending until the system is phased out of the inventory. It overlaps the production phase. (AFR 800-2) [1]

DERIVES - This relationship indicates that a function on the path derives either external information (external output) or internal system information (internal output) as part of its activities. (See also Information Flow)

Design and Construction - Minimum or essential requirements that are not controlled by performance characteristics, interface requirements, or referenced documents shall be specified. They shall include appropriate design standards, requirements governing the use or selection of materials, parts and processes, interchangeability requirements, safety requirements, and the like. Requirements for materials to be used in the item or service covered by the specification shall be stated, except where it is more practicable to include the information in other paragraphs. Requirements of a general nature should be first, followed by specific requirements for the material. Definitive documents shall be referenced for the material when such documents cover materials of the required quality. [3]

Design Engineering - This function uses the technical information (requirements, goals, criteria, constraints, etc.) developed through the systems engineering process to develop detailed design approaches, design solutions, and the test procedures to prove these solutions. [6] (See also Engineering Management)

Design Requirements - The minimum or essential design and construction requirements which are not addressed by other constraint requirement types: performance, physical, operability, and test requirements. During the initial phases of systems requirements engineering, certain design and construction standards (see Design and Construction) may be specified directly or by reference to other specifications or standards. As the system development continues, engineering analysis and trade study results (as well as other engineering activities such as prototyping and simulations) may indicate the need for additional design constraints which are practicable and necessary for the system's operation and maintenance (O&M).

Development (Part I or Type B5) Specification - A document which specifies the requirements peculiar to the design, development, functional performance, test, and qualification of the configuration item. It establishes performance criteria and test criteria for which the program shall be designed/ developed [MIL-STD-483(USAF)]. [7] (See also Type B Specification and Specifications)

Development Test & Evaluation (DT&E) - That testing and evaluation of individual components, subsystems, and, in certain cases, the complete system, which is conducted predominantly by the contractor. [7]

Discrete Event Simulation - On the system level, a discrete event simulation may be utilized to support computer system studies. A discrete event simulation is one in which information blocks and computer program timing can be replicated allowing evaluation of throughput capability and identification of potential design problems. This type of simulation is used to check the software design for possible discrepancies that might cause the system to be saturated as a result of either information overloads or time responses that are slower than required. These studies provide estimates of computer sizing and timing for the processing requirements and they evaluate the real-time computational conflicts, including the effects of interrupts. [9] (see also functional simulation, Scientific Simulation, Engineering Simulation)

Electromagnetic Compatibility (EMC) - Defined as "the capability of an equipment, component, subsystem or system to operate in its operational electromagnetic environment at design levels of efficiency, without causing or suffering unacceptable degradation due to electromagnetic interference." The application of approved EMC standards in the development and procurement of equipment is required by AFR 80-23 (para 6d). [1] Where applicable, requirements pertaining to electromagnetic radiation shall be stated in terms of the environment which the item must accept and the environment which it generates. [3]

Electronic Warfare (EW) - The mission capability of Command, Control & Communications systems is continually threatened by the possibility of electronic countermeasures (ECM) such as spoofing and jamming. Potential adversaries put a high emphasis on ECM and have a constantly improving ECM technology base. To be responsive, each Command, Control & Communications system concept must have as little potential for ECM exploitation as possible, electronic counter-counter measure (ECCM) technology base must be vigorous, and incorporation of ECCM into systems must be timely. [1]

Engineering Change - An alteration in the configuration item or items, delivered, to be delivered, or under development, after formal establishment of its configuration identification. [4]

Engineering Change Proposal (ECP) - A term which includes both a proposed engineering change and the documentation by which the change is described and suggested. [4]

Engineering Management - The management of the engineering and technical effort required to transform a military requirement into an operational system. It includes the system engineering required to define the system performance parameters and prefered system configuration to satisfy the requirement, the planning and control of technical program tasks, integration of the engineering specialties, and the management of a totally integrated effort of design engineering, specialty engineering, test

engineering, logistics engineering, and production engineering to meet cost, technical performance and schedule objectives. The engineering management task of the government program office assures that the technical functions in the program office are properly planned and implemented, and that the technical functions performed under contract are tailored, monitored, and controlled to best meet the needs of the system or program. These functions (together with certain supporting functions) are: Systems Engineering (including Requirements Engineering), Design Engineering, Specialty Engineering, Test Engineering, Production Engineering, Logistics Engineering, Civil Engineering, Human Factors Engineering, Configuration Management, Technical Data Control, and Technical Program Planning and Control. [10]

Engineering Simulation - Engineering simulation is a further refinement of the scientific simulation in which the final software design is evaluated by driving this software with realistic input data generated from representative scenarios. These simulations, executed on a general purpose computer, are characteristic of the types of tools needed in system and software requirements definition and evaluation. [9] (See also functional simulation, discrete event simulation, scientific simulation)

Environmental Conditions - Environments that the system or equipment is expected to experience in shipment, storage, service, and use. The following subjects should be considered for coverage: natural environment (wind, rain, temperature, etc.); induced environment (motion, shock, noise, etc.), electromagnetic signal environment; shipboard magnetic environment; and environmental conditions due to enemy action (over-pressure, blast, underwater explosions, radiation, etc.).

External Interface - (Also called Intra-System Interface). The interfaces between the system being specified and other systems with which it must be compatible. [3] (See also Interface)

Formal Qualification Tests (FQT) - A formal test conducted in accordance with the Air Force-approved test plans and designed to be a complete and comprehensive test of the CPCI prior to FCA. It is conducted after the design process culminates (AFR 80-14, Vol. II). [7]

Full-Scale Development Phase (FSD) - The period when the system/equipment and the principal items necessary for its support are designed, fabricated, tested, and evaluated. The intended output is, as a minimum, a preproduction system which closely approximates the final product, the documentation necessary to enter the production phase, and the test results which demonstrate that the production product will meet stated requirements. (AFR 800-2) [1] (see also Acquisition Life Cycle)

Function (Functional Requirement Set, Functional Requirements) - A function is a discrete activity within a system. The functional requirements represent the total discrete system activities required to achieve a specific objective, this is most often referred to as the mission objective. A functional requirement identifies what must be accomplished without identifying any aspect concerning the means such as hardware,

computer programs, personnel, facilities, or procedural data. Functional requirements represent a problem statement devoid of any overtones or specifics regarding real or conceptual solutions which satisfy any or part of the needed functions.

Note 1: Functions take on different meanings within the three types of system documentation as required by MIL-STD-483 (USAF). Type A specification functions are defined for the system as a whole as defined above. Type B5 specifications define CPCI function to include the inputs, processing, and outputs. The Computer Program Components (CPCs) of the Type C5 specification may correspond to the functions in the Type B5 specification, if the B5 requirements satisfy the computer program developer's design approach. (See [11], para. 4.3.1 and Appendix A4)

For the purpose of requirements engineering, functions are defined to be the same as Type A Specification functions. In documenting functions in Type B5 specifications, the associated inputs and outputs are included.

Note 2: The revised AFR 57-1 provides a slightly different definition of a function: The action for which a system or equipment item is specially fitted or used. [13]

Functional Analysis - System functions and sub-functions shall be progressively identified and analyzed as the basis for identifying alternatives for meeting system requirements. System functions as used above include the mission, test, production, deployment, and support functions. All contractually specified modes of operational usage and support shall be considered in the analysis. System functions and sub-functions shall be developed in an iterative process based on the results of the mission analysis, the derived system performance requirements, and the synthesis of lower-level system elements. Performance requirements shall be established for each function and sub-function identified. When time is critical to a performance requirement, a time line analysis shall be made. [10] (See also Systems Engineering)

Functional Area - A distinct group of system performance requirements which, together with all other such groupings, forms the next lower level breakdown of the system on the basis of function. [4] The second level in the functional hierarchical structure. (See also System Segment, CI and CPCI)

Functional Characteristics - Quantitative performance, operating and logistic parameters and their respective tolerances. Functional characteristics include all performance parameters, such as range, speed, lethality, reliability, maintainability, and safety. (DOD Directive 5010.19) [4]

Functional Hierarchical Structure - This form of organization is suited for structuring system functional requirements in a logical arrangement of subordinate discrete activities which must be performed. The functions of the system are grouped into higher levels of organization representing the

first possible breakout of the system. Upper-level functions are refined by the identification of subordinate levels. Each level of the hierarchy is limited to six functions or less. (See also System Segment, Functional Area, Configuration Item, Computer Program Configuration Item)

Functional Performance - The ability of the software to satisfy its mission requirements as allocated from the System Specification and as contractually specified in the Development Specification. [2]

Functional Requirements - see Function

Functional Simulation - A functional simulation generally consists of a set of building blocks which functionally define the basic elements of the system such as the sensor models, aircraft dynamics, navigation, weapon delivery, and the environment. This type of simulation is used to analyze performance in support of system requirements definition. To support this analysis activity, the simulation may be utilized to generate mission scenarios in order to evaluate system performance parameters and tradeoff studies associated with various system elements, such as the sensors, etc. [9] (See also discrete event simulation, scientific simulation, engineering simulation)

Government Furnished Property (GFP) - Contracts may require the use of GFP, either as end item design requirement or as a part of the system. In such cases, a schedule is included in the contract for delivery of the GFP to the contractor at a date permitting his evaluation for serviceability before it is needed for installation. Engineering data on the GFP must be provided at a date which permits the contractor's engineers to incorporate it, or the interface with it, into the design of the system. [1]

Human Engineering - Human Engineering is usually a contractor design and review process that interacts with other processes such as mission requirements analysis, functional analysis and requirement allocation, the development of workspace mockups, equipment detail design, test and evaluation, etc. (MIL-H-46855A applies.) The contractor is tasked to identify and investigate areas where interactions of human performance and other elements of the system are critical to the system-effectiveness. The contractor's end task is to translate controller/situation, human/information and man/machine functional interface requirements into human engineering design criteria for incorporation into system, equipment, software and facility specifications and delivered products. [1] (See also Human Factors Engineering)

Human engineering requirements for the system/item should be described in specifications and applicable documents (e.g., MIL-STD-1472) included by reference. The specifications should also specify any special or unique requirements, e.g., constraints on allocation of functions to personnel, and communications and personnel/equipment interactions. Included, should be those specified areas, stations, or equipment that require concentrated human engineering attention due to the sensitivity of the operation or criticality of the task, i.e, those areas where the effects of human error would be particularly serious. [3]

Interfaces between software and the user should be specified in the Development (Part I) Specification. Inputs and outputs should be self explanatory, easy to learn and understand, unambiguous, and designed to avoid misinterpretation. [2]

Human Factors Engineering - This function is a part of the mainstream engineering effort throughout the system life cycle. It uses data from, and contributes to, the system engineering process in developing a best mix of specification requirements. Its objective is to ensure that the human component of the system can safely and effectively operate, maintain, support, and control the system in its intended operational environment. It is also concerned with providing engineering data for use in hardware, software, or people cost-effective trade studies, and with developing plans for training and training equipment (see AFR 800-15). [6] (See also Engineering Management and Human Engineering)

Implementing Command - The command or agency designated by Program Management Directive (PMD) as responsible to achieve the program objectives or program phase objectives established in the PMD. (Ref: AFR 800-2) [13]

The Air Force command responsible for the acquisition of the system (subsystem or item). The procuring activity is usually resident within the Implementing Command. Program management responsibility normally is transferred to the designated supporting command according to a predetermined agreement. Similarly, the responsibility of system operation and maintenance is turned over to the using command. [8]

Information Flow - The description of the flow of information into, within, and out of the system. The information flow builds upon the I/O hierarchical structure by providing a means of analyzing the system as an information processing system. During this analysis, the flow relationships between external system inputs and resulting outputs are identified. This method permits the various relationships between associated functions and the internal information necessary to support the derivation of the output to be identified. The flow associations between system information are described using the information-flow relationships: USES, DERIVES, UPDATES, PROVIDES, and RECEIVES. The informational flow indicates only the relationship between system functions, system information (external and internal system I/O), and using activities (organizations, operational units, or positions) and does not imply any lapse in time or intermediate I/O being used, derived, or updated.

Initial Operational Capability (IOC). The first attainment of the capability to employ effectively a weapon, item of equipment, or system of approved specific characteristics, and which is manned or operated by an adequately trained, equipped, and supported military unit or force. (Source: JCS Pub. 1) [13]

I/O Hierarchical Structure - The logical hierarchical description of the discrete system inputs and outputs (external I/O) and the internal information requirements necessary for the system's operation. The emphasis

on the I/O structure is to arrange the information requirements into structures by breaking the information into logical subordinate parts or simply as groupings of information. The well-organized structure is effective in communicating the I/O requirements and for identifying missing I/O requirements.

Interface - The functional and physical characteristics required to exist at a common boundary between two or more equipments/computer programs. Interfaces between equipment/computer programs provided by different developing agencies (contractors), or between development items and government furnished property or external systems, require explicit documentation. [8] (See also External Interface and Internal Interface)

Life Cycle Cost (LCC). The total cost of an item or system over its full life. It includes the cost of acquisition, ownership (operation, maintenance, support, etc.) and, where applicable, disposal. To be meaningful, an expression of life cycle cost must be placed in context with the cost elements included, period of time covered, assumptions and conditions applied, and whether it is intended as a relative comparison or absolute expression of expected cost effects. (Source: AFR 800-11) [13]

Internal Interface (also called Inter-System Interface) - The interfaces between and within the system being specified (e.g., between system segments, functional areas, configuration items) [3] (See also Interface)

Life Cycle Cost Analysis - Life Cycle Cost Analysis is performed by the contractor periodically throughout the acquisition to access the cost of acquisition and ownership. This effort results in an identification of the economic consequences of system design alternatives. [10] (See also Systems Engineering)

Logical Organizational Relationships - Logical organizational relationships are shown by structuring the discrete functions and the information requirements (external and internal input/output) of the system into hierarchical structures: Functional Hierarchical Structure, and I/O Hierarchical Structure.

Logistics Engineering - This function provides inputs to the systems engineering process in all acquisition phases. In general, these inputs are the support environment descriptors and constraints. This function uses the technical data developed by the systems engineering process to refine the support plans, concepts, and requirements for system support in the deployment phase and in operational utilization. The logistics engineering function is a part of the mainstream engineering effort to develop and achieve a supportable and cost-effective system. This function uses the detailed drawings which are prepared by design engineering to develop the specific support requirements; that is, to develop such specific support items as tools, test equipment, personnel skills, and maintenance procedures. (For other information concerning logistics engineering responsibilities, see AFR 800-8 and AFP 800-7.) [6] (See also Engineering Management)

Logistics Support Analyses - The contractor is usually tasked to conduct logistic support analyses leading to the definition of support needs (e.g., maintenance equipment, personnel, spares, repair parts, technical orders, manuals, transportation and handling, etc.). These analyses address all levels of operations and maintenance and results in requirements for support. [10] (See also Systems Engineering)

Maintainability - Closely related and inseparable from Reliability is the specialty, Maintainability. Maintainability is a characteristic of the design and installation expressed as the probability that an item will be restored to a specified condition within a given period of time when the maintenance is performed using prescribed procedures and resources. (See also Reliability and Availability) [1] The revised AFR 57-1 emphasizes the following definition: a measure of the time or maintenance resources needed to keep an item operating or restore it to operational (or in the case of certain munitions, serviceable) status. Maintainability may be expressed as the time to do maintenance, as the total required manpower, or as the time to restore a system to operational (or serviceable) status. (Source: AFR 80-5) [13]

Numerical maintainability requirements shall be stated in such terms as mean-time-to-repair (MTTR) or maintenance man-hours per flight/operational hour. Determination of realistic requirements is necessary. Qualitative requirements for accessibility, modular construction, test points, and other design requirements may be specified as required. [3]

Specifications shall specify the quantitative maintainability requirements. The requirements shall apply to maintenance in the planned maintenance and support environment and shall be stated in quantitative terms. Examples are:

a. Time (e.g., mean and maximum downtime, reaction time, turnaround time, mean and maximum times to repair, mean time between maintenance actions).

b. Rate (e.g., maintenance manhours per flying hour, maintenance manhours per specific maintenance action, operational ready rate, maintenance hours per operating hour, frequency of preventive maintenance).

c. Maintenance complexity (e.g., number of people and skill levels, variety of support equipment).

d. Maintenance action indices (e.g., maintenance costs per operating hour, manhours per overhaul). [3]

Maintainability as applied to software is specification, design, and development of code in a manner which facilitates the task of modification to correct deficiencies and to satisfy new or changing requirements. A potential source of confusion exists regarding subtle distinctions between the hardware and software definition of maintainability. Hardware maintenance is the restoration of hardware to its original design, whereas software maintenance is defined as both error correction and modification of the original design (both of which imply change rather than restoration)

Since there is little chance that the usage of either set of definitions will be discontinued, the procuring agency should bear these differences in mind when participating in the establishment of maintainability criteria for the total system. Software maintenance features in terms of growth requirements may be specified in the Development (Part I) Specification. Additional features such as modularity should be requested in the RFP, responded to in the CPDP, and implemented by the contractor in the design, and reflected in the Product (Part II) Specification. [2]

Maintenance Concept. A description of maintenance considerations and constraints. A preliminary maintenance concept is developed and submitted as part of the preliminary operational concept for each alternative solution candidate by the operating command with the assistance of the implementing and supporting commands. The preliminary maintenance concept is refined during the demonstration and validation phase to become the system maintenance concept during full scale engineering development (FSED). During FSED, the system maintenance concept is expanded in scope and detail and removed from the system operational concept to become the maintenance plan. (Source: AFR 66-14) [13]

Milestone Zero Decision. The program initiation decision by competent authority that valid mission need exists and alternative solutions should be systematically and progressively identified and explored. Secretary of Defense approval of the need is required to initiate major system acquistion programs. Secretary of the Air Force approval is required to initiate Air Force designated acquisition programs (AFDAP). HQ USAF approval by PMD is required to initiate all other acquisition programs. [13]

Mission Area. A segment of the defense mission as established by the Secretary of Defense. (Source: AFR 800-2) [13]

Mission Area Analyses. Continuous analysis of assigned mission responsibilities in the several mission areas to identify deficiencies in the current and projected capabilities to meet essential mission needs and to identify opportunities for the enhancement of capability through more effective systems and less costly methods. Missions area analysis should conform with short, mid, and long range planning guidance. The objectives of mission area analysis are to identify capability deficiencies and assess the relative values of operational needs. [13]

Mission Area Planning. A continuous HQ USAF and command planning activity which directs and coordinates mission area analysis and uses the product of that analysis to help make program, budget, modification and acquisition, force structure, strategy and tactics decisions. [13]

Mission Element. A segment of a mission area critical to the accomplishment of the mission area objectives and corresponding to a recommendation for a major system or designated non-major system capability as determined by the Air Force. (Ref: AFR 800-2) [13]

Mission Element Need Analysis (MENA). A mandatory attachment of the SON which cites the command mission and tasks, documents of the salient results of the mission analysis which identified the operational deficiency, states command needs for mission task performance, and provides constraints on acceptable solutions. [13]

Mission Element Need Statement (MENS). A statement prepared by HQ USAF to identify and support the need for a new or improved mission capability. It is normally based on one or more SONs. The mission need may result from a projected deficiency or obsolescence in existing systems, a technological opportunity, or an opportunity to reduce operating cost. The MENS is submitted to the SECDEF or SAF as appropriate for a Milestone 0 decision. (Ref: DOD Directive 5000.2) [13]

Mission Reliability. A measure of the ability of a system to complete its planned mission or function. Mission reliability may be expressed as Mission Completion Success Probability (MCSP), Mean Mission Duration (MMD), or as Mean Time Between Critical Failure (MTBCF) as appropriate. (Source: AFR 80-5) [13]

Mission Requirements Analysis - Impacts of the stated system operational characteristics, mission objectives, threat, environmental factors, minimum acceptable system functional requirements, technical performance, and system figure(s) of merit as stipulated, proposed, or directed for change are analysed during the conduct of the contract. These impacts are examined continually for validity, consistency, desirability, and attainability with respect to current technology, physical resources, human performance capabilities, life cycle costs, or other limitations. The output of this analysis will either verify the existing requirements or develop new requirements which are more appropriate for the mission. [10] (See also Systems Engineering and System Capability requirements)

Operability. (Sometimes called System-Effectiveness or System Operational Effectiveness) - Operability includes system availability and dependability. Availability incorporates the aspects of reliability and maintainability, dependability incorporates the aspects of survivability and vulnerability (S/V). Each of these operability categories may be influenced by design related issues, policy related impact, or non-controllable factors.

Operating Command. The command or agency primarily responsible for the operational employment of a system, subsystem or item of equipment. The operating command usually submits the SON. The operating command is a participating command. (Ref: AFM 11-1, Vol I) [13]

Operational Concept. A statement about intended employment of forces that provides guidance for posturing and supporting combat forces. Standards are specified for deployment, organization, basing, and support from which detailed resource requirements and implementing programs can be derived. (Source: (AFM 11-1, Vol I) [13]

Operational Need or Mission Need. A capability to successfully perform one or more mission tasks or functions required to achieve mission or mission area objectives. The operational need is expressed in terms of task and functional capabilities (what must be done and how well) not in terms of specific hardware or software system characteristics. [13]

Operational Test and Evaluation (OT&E) - OT&E is conducted to estimate the system's military utility, operational effectiveness, operational suitability, and to identify deficiencies. In addition, OT&E provides information on organization, personnel requirements, doctrine and tactics, also it may provide data to support or verify material in operating instructions, publications and handbooks.

OT&E normally composed of initial OT&E (IOT&E) and follow-on OT&E (FOT&E) is essentially an operational analysis of a system's performance where the complete system is tested and evaluated against operational criteria by personnel with the same qualifications as those who will use, maintain and support the system when deployed. There will be situations when OT&E requirements will be satisfied by the formal DT&E or by combining OT&E activities with DT&E. [1]

The revised AFR 57-1 provides the following definition: OT & E - test and evaluation conducted to estimate the system's military utility, operational effectiveness and operational suitability. (Source: AFR 800-2) [13]

OR - Any one of the alternate paths may lead to the next activity. The conditions upon which the alternate paths are selected are associated with the OR. (See also Control-Flow)

Participating Command. A command or agency designated by HQ USAF to support and advise the Program Manager (PM). A supporting command is a participating command. (Source: AFR 800-2) [13]

Performance Requirements (also called Performance Characteristics [3]) - Performance requirements identify "how well" the functions of the system must be accomplished. The performance requirements are the essential quantifiable statistical parameters upon which the successful accomplishment of system functions can be evaluated, such as timeliness and accuracy. The timing performance constraints include computational-solving times, count-down or event timing, and timing allocations as established through engineering analysis.

Personnel and Training - Where applicable, requirements imposed by or limited by personnel or training considerations shall be specified in development specifications. Training considerations shall include existing facilities, equipment, special/emergency procedures (associated with hazardous tasks) and training simulators, as well as the need for additional facilities, equipment, and simulators. [3]

Manpower requirements may be taken care of for projects by the User/Operating Command manpower agency. The number of O&M personnel, their AFSC(s) and skill level(s), may be included in the program direction,

influencing the system/equipment design. On the other hand, the manpower agency may request program office support in determining the appropriate manning for a new or complex system. In this case the program office can task the contractor to perform studies for determining the manpower requirements. [1]

Physical Characteristics - Quantitative and qualitative expressions of material features, such as composition, dimensions, finishes, form, fit, and their respective tolerances (DOD Directive 5010.19). [4] These characteristics in a development, product or material specification shall set forth requirements such as weight limits, dimensional limits, etc., necessary to assure physical compatibility with other elements and not determined by other design and construction features or referenced drawings. They shall also include considerations such as transportation and storage requirements, security criteria, durability factors, health and safety criteria, command control requirements, and vulnerability factors. [3] (See also Physical Requirements)

Physical Requirements - Physical requirements are those requirements which constrain or significantly influence the design solution in a physical manner. The physical constraints include power, physical features (size and weight), environmental considerations (controlled or natural), human performance capabilities and limitations (human factors), predetermined internal system interfaces (inter-system interfaces) and external system interfacing (intra-system interfaces), use of existing equipment (off-the shelf) and Government Furnished Property (GFP), and use of standard parts. (See also Physical Characteristics)

Preliminary Qualification Tests (PQT) - A formal test conducted in accordance with Air Force-approved test plans and designed to be an incremental process which provides visibility and control of the computer program development during the time period between CDR and FQT. A PQT should be conducted for those functions which are critical to the CPCI (AFR 800-14, Vol. II). [7]

Procuring Activity (Also called Procuring Agency) - The collection of administrative, management and technical expertise which is organized under a program manager directly responsible for the acquisition of a system. The term System Program Office (SPO) is used in the Electronic Systems Division (ESD) of AFSC to designate a procuring activity responsible for a large system acquisition. [8] (See also Program Office and Implementing Command)

Production Engineering - This function uses the technical data developed through the systems engineering process to develop the plans and procedures for tooling, materials, quality assurance, and manufacturing. The production engineering function is a part of the mainstream engineering effort to develop and achieve producible and cost-effective design solutions. (For other information concerning production engineering responsibilities, see AFR 800-9) [6] (See also Engineering Management)

Production Engineering Analysis - Production engineering analysis is an integral part of the system engineering process. It includes producibility analyses, production engineering inputs to system effectiveness, trade-off studies, and life cycle cost analyses and the consideration of the materials, tools, test equipment, facilities, personnel, and procedures which support manufacturing in RDT&E and production. Critical or special producibility requirements are identified as early as possible and are an input to the program risk analysis. Where critical or special production engineering requirements limit the design, these requirements are included in applicable specifications. Long lead time items, material limitations, transition from development to production, special processes, and manufacturing limitations are considered and documented during the system engineering process. The contractor identifies and takes necessary steps to reduce high-risk manufacturing areas as early as possible. [10] (See also Systems Engineering)

Production Phase - The period from production approval until the last system/ equipment is delivered and accepted. The objective is to efficiently produce and deliver effective and supportable systems to the operating units. It includes the production and deployment of all principal and support equipment. (AFR 800-20 [1])

Product Specification - A document or series of documents which contain the detailed technical description of the CPCI as designed and coded. It is a complete description of all routines, limits, timing, flow, and data base characteristics of the computer program, limits, timing, flow, and data coded instructions. Equivalent to "Part II CPCI specification" or "Type C5 Specification". [7] (See also Type C Specification and Specifications)

Program Management Directive (PMD) - The official HQ USAF management directive used to provide direction to the implementing and participating commands and satisfy documentation requirements. It will be used during the entire acquisition cycle to state requirements and request studies as well as initiate, approve, change, transition, modify or terminate programs. The content of the PMD, including the required HQ USAF review and approval actions, is tailored to the needs of each individual program. (AFR 800-2) [1]

Program Management Plan (PMP) - The document developed and issued by the Program Manager which shows the integrated time-phased tasks and resources required to complete the task specified in the PMD. It defines the support required from all participating organizations, is tailored to the needs of each individual program, and contains only that information deemed necessary by the program manager. (AFR 800-2) [1]

Program Office (PO) - The field office organized by the program manager to assist him in accomplishing the program tasks. (AFR 800-2) (See also Procuring Activity) [1]

PROVIDES - This relationship indicates that a using activity is the source of the external output. (See also Information Flow)

Quality Requirements. The term 'quality requirements' denotes system requirements which are complete, consistent, testable, and traceable. This characteristic is the result of the requirements being discretely identified and well-organized. (see also Requirements Engineering)

RECEIVES - This relationship indicates that a using activity is the recipient of the external output. (See also Information Flow)

Reliability - As defined in AF Regulation 80-5, Reliability and Maintainability Programs for Systems, Sybsystems, Equipment, and Munitions, Reliability is the probability that a part, components, subassembly, assembly, subsystem or system will perform for a specified interval under stated conditions with no malfunction or degradations that require corrective maintenance actions. Hardware reliability may also be expressed in terms such as Mean Time Between Failure (MTBF) or Mean Time Between Maintenance Action. [1]

Reliability requirements shall be stated numerically with confidence levels, as appropriate, in terms of mission success or hardware mean time between failures. Initially, reliability may be stated as a goal and a lower minimum acceptable requirement. During contract definition, or equivalent period, realistic requirements shall be determined and incorporated in the specification with requirements for demonstration. Reliability requirements shall never be stated as a goal in Type C (product) specifications. [3]

Reliability is a difficult and perhaps inappropriate term when applied to software because this item has an entirely different meaning for hardware. Since a computer program never wears out it is virtually impossible to predict or analyze failure rates. Any failure of the computer program is a latent design deficiency and its occurrence cannot be adequately predicted. In this respect a computer program cannot be designed for reliability and cannot be tested or evaluated for reliability. Reliability should not apply to computer programs as end items although the computer programs may be used to enhance system reliability. [2] (See also Availability and Maintainability)

Required Operational Capability (ROC) - The ROC identifies the need for a new or improved operational capability. The formal numbered document used under previous editions of AFR 57-1, (27 Nov 1963 through 31 Aug 1977) to identify an operational need and to request a new or improved capability for the operating forces. [13] Once the ROC is validated by HQs USAF, the PMD, which authorizes AFSC to establish a Program Office cadre, is issued. [2]

Requirements Allocation - Each function and sub-function shall be allocated a set of constraint requirements. These requirements shall be derived concurrently with the development of functions, time-line analyses, synthesis of system design, and evaluation performed through trade-off studies and system/ cost effectiveness analysis. Time requirements which are prerequisites for a function or set of functions affecting mission success, safety, and availability shall be derived. The derived

requirements shall be stated in sufficient detail for allocation to hardware, computer programs, procedural data, facilities, and personnel. When necessary, special skills or peculiar requirements will be identified. Allocated requirements shall be traceable through the analysis by which they were derived to the system requirement they are designed to fulfill. [10] (See also Systems Engineering)

Requirements Analysis - (See Requirements Engineering)

Requirements Definition - (See Requirements Engineering)

Requirements Engineering - An iterative process of defining the system requirements and analyzing the integrity of the requirements. This process involves all areas of system development preceding the actual design of the system. The products of the requirements engineering process can be evaluated for completeness, consistency, testability, and traceability. The essential goal of requirements engineering is to thoroughly evaluate the needs which the system must satisfy. (See also Engineering Management)

Requirement Types - See System Requirements

Requirements Traceability - See Traceability

Safety - Requirements for system safety are described to preclude or limit hazard to personnel, equipment, or both. To the extent practicable, these requirements are imposed by citing established and recognized standards. Limiting safety characteristics peculiar to the item due to hazards in assembly, disassembly, test, transport, storage, operation or maintenance are stated when covered neither by standard industrial or service practices nor the system specification. "Fail-safe" and emergency operating restrictions are included when applicable. These include interlocks and emergency and standby circuits required either to prevent injury or provide for recovery of the item in the event of failure. [3] (See also System Safety)

Scientific Simulation - Scientific simulation is the primary simulation used in detailed computer program requirements definition and algorithm design. Scientific simulation consists of a functional simulation (for example, FORTRAN version) of the proposed end-item software, interfaced with simulations representing sensor and environmental models. Such a scientific simulation allows the study of the major end-item software, and provides further information to be used for system performance evaluation. [9] (See functional simulation, discrete event simulation, engineering simulation)

Segment - (See System Segment)

Simulation - See Functional Simulation, Discrete Event Simulation, Scientific Simulation, Engineering Simulation.

Software - Software denotes computer programs and computer data. A computer program is a series of instructions or statements in a form

acceptable to a computer, designed to cause the computer to execute an operation or operations. Computer programs include operating systems, assemblers, compilers, interpreters, data maintenance/diagnostic programs, as well as applications programs such as payroll, inventory control, operational flight, strategic, tactical automatic test, crew simulator, and engineering analysis. Computer programs may be either machine-dependent or machine-independent, and may be general-purpose in nature or be designed to satisfy the requirements of a specialized process or particular users. Computer data is a collection of data in a form capable of being processed and operated on by a computer, such as a data base, or analog or digital inputs to a computer program that are necessary for its operation. [2], [8] (See also Computer Program)

Speciality Engineering - This term refers to the engineering efforts of reliability, maintainability, safety, survivability, vulnerability, corrosion prevention, structural integrity, etc. These engineering functions are part of the mainstream engineering effort to develop a best mix of specification requirements and achieve cost-effective design solutions. [6] (See also Engineering Management)

Specification (See also Systems Engineering) - A document intended primarily for use in procurement, which clearly and accurately describes the essential technical requirements for items, materials or services including the procedures by which it will be determined that the requirements have been met. (DOD Directive 4120.3) [4] MIL-STD-490 and MIL-STD-483 Specification types are:

System specification. A document which states the technical and mission requirements for a system as an entity, allocates requirements to functional areas (or configuration items), and defines the interfaces between or among the functional areas. (See also Type A) [4]

Development specification. A document applicable to an item below the system level which states performance, interface, and other technical requirements in sufficient detail to permit design, engineering for Service use, and evaluation. (see also Type B) [4]

Product specification. A document applicable to a production item below the system level which states item characteristics in a manner suitable for procurement, production and acceptance. (See also Type C) [4]

Statement of Operational Need (SON). A formal numbered document used to identify an operational deficiency and state the need for a new or improved capability for USAF forces. Operational needs are based on short term and long term capability objectives and may result from a projected deficiency or obsolescence in existing capabilities, a technological opportunity, or an opportunity to reduce operating/support cost. It usually begins the system acquisition process and is normally followed by the conceptual phase, however, any appropriate phase may follow. Satisfying a SON will normally require a combination of research, development, test, modification, or acquisition efforts that will enhance USAF forces' capabilities. [13]

Supporting Command - A command providing direct support to a system or test program. Examples include the Air Force Logistics Command (AFLC) and the Air Training Command (ATC). See also implementing command and using command. [8] The revised AFR 57-1 provides the following definition: The command assigned responsibility for providing logistics support; it assumes program management responsibilty from the implementing command. The supporting command is a participating command. (Ref: AFR 800-2) [13]

Synthesis - Sufficient preliminary design is accomplished to confirm and assure completeness of the performance and design requirements allocated for detail design. The performance, configuration, and arrangement of a chosen system and its elements and the technique for their test, support, and operation are portrayed in a suitable form such as a set of schematic diagrams, physical and mathematical models, computer simulations, layouts, detailed drawings, and similar engineering graphics. These portrayals shall illustrate intra- and inter-system and item interfaces, permit traceability between the elements at various levels of system detail, and provide means for complete and comprehensive change control. This portrayal is the basic source of data for developing, updating, and completing (a) the system, configuration item, and critical item specifications, (b) interfacing control documentation; (c) consolidated facility requirements; (d) content of procedural handbooks, placards, and similar forms of instructional data; (e) task loading of personnel; (f) operational computer programs; (g) specification trees; and (h) dependent elements of work breakdown structures. [10] (See) also Systems Engineering)

System - A composite of items, assemblies (or sets), skills, and techniques capable of performing and/or supporting an operational (or non-operational) role. A complete system includes related facilities, items, material, services, and personnel required for its operation to the degree that it can be considered a self-sufficient item in its intended operational (or non-operational) and/or support environment. (AFR 65-3) [1],[8],[4]

System Acquisition Process. A sequence of specified decision events and phases of activity directed to achievement of established program objectives in the acquisition of Defense systems and extending from approval of a mission need through successful deployment of the Defense system or termination of the program. (Source: AFR 800-2) [13]

System/Acquisition Life Cycle - Normally, it consists of five phases (Conceptual, Validation, Full-Scale Development, Production, and Deployment) with key decision points between each of the first three phases (Program, Ratification, and Production Decisions). A program may skip a phase or have program elements in any or all other phases. (See AFR 800-2 and AFSCP 800-3) (See also Acquisition Life Cycle) [1]

System Capability Requirements - The mission oriented needs which the system must perform to satisfy the requirements of the using agency. (See also Mission Requirements Analysis)

System/Cost Effectiveness Analysis - A continuing system/cost effectiveness analysis insures that engineering decisions, resulting from the review of

alternatives, are made only after considering their impact on system effectiveness and cost of acquisition and ownership. The contractor is tasked to identify alternatives which would provide significantly different system effectiveness or costs than those based upon contract requirements. [10]

System Design Concept. An idea expressed in terms of general performance, capabilities, and characteristics of hardware and software oriented either to operate or to be operated as an integral whole in meeting a mission need. (Source: OMB Circular A-109) [13]

Systems Engineering - The application of scientific and engineering efforts to transform an operational need or statement of deficiency into a description of system requirements and a preferred system configuration that has been optimized from a life cycle cost viewpoint. The process of systems engineering has three principal elements: functional analysis, synthesis, and trade studies or cost-effectivess optimization. The process uses a sequential and iterative methodology to reach cost-effectivess solutions. The technical information developed in this process is used to plan and integrate the engineering effort for the system as a whole, during the definition, design, test and evalution, production, deployment, support, and modification of a system or equipment item. (AFR 800-3) [1] (See also Engineering Management)

System engineering for the total system or a functional area (system element or segment) is normally vested in a single contractor or Government agency. System engineering as it relates to configuration management, is the application of scientific and engineering efforts to transform an operational need into a description of system performance parameters and a system configuration must be ultimately called out in the CI specifications. In this way, the system engineering agency or contractor generates requirements for configurations which will satisfy the operational need, constrained technically only by the content of the system specification. The system engineering agency or contractor is responsible for assessing the impact of changes to CI specifications or to the system specification. This includes modifications to operational systems. (See MIL-STD-490 for system engineering criteria.) [1]

The following typical tasks are conducted (as appropriate) in performing system engineering (see separate definitions for each):

Mission Requirements Analysis
Functional Analysis
Requirements Allocation
Synthesis
Logistics Support Analysis
Life Cycle Cost Analysis
Trade-Off Studies
Production Engineering Analysis
Specifications [10]

System Engineering Management Plan (SEMP) - A contractor's proposal describing this approach to system engineering management to be applied in a specific acquisition contract. The SEMP normally consists of three major parts: (1) System Engineering, (2) Technical program planning and control, and (3) Engineering integration. (MIL-STD-499A) [3,5,8]

System Flow Relationships - System flow relationships can be shown be organizing the discrete requirements in terms of control flow and ___ information flow.

System Requirements - System Functions and Constraints

System Safety - Defined by MIL-STD-882 to be the optimum degree of safety within the limits of operational effectiveness, time and cost, attained through specific application of system safety management and engineering principles throughout all phases of a system's life cycle. It is very important to realize that system safety is concerned with the safety of both personnel and equipment. The application of this discipline to ensure the preservation of equipment immediately expands its scope beyond that of the traditional safety field, and establishes it as an engineering area. As implied above, the basic guidance document for system safety is MIL-STD-882, System Safety Program for Systems and Associated Subsystems and Equipment: Requirements for. This is a very broad document and must be tailored to fit the individual program. The other basic document is AFR 127-8, Responsibilities for USAF System Safety Engineering Programs, and the AFSC supplement thereto. This gives specific requirements to be applied to most programs. [1] (See also Safety)

Systems Operational Concept (SOC) - A formal document that describes the intended purpose, employment, deployment, and support of a system. It assists in identifying the variables associated with satisfying the operational need and provides initial guidance to operating forces for employing the new or improved system. It provides information for posturing combat forces and specifies standards for deployment, organization, basing and support from which detailed resource requirements and implementing programs can be derived. It must be compatible with long range Air Force goals and objectives and consistent with Air Force strategy, force structure, concepts for the future employment of aerospace forces, and current and emerging doctrin. Prior to FSED, it contains as an integral part, the maintenance concept prepared per AFR 66-14. [13]

System Segment - A discrete package of system performance requirements, functional interfaces and configuration items allocated to one developing agency directly responsible to the procuring activity for that part of the system's total performance. The term "system segment" can be synonymous with "subsystem" or "functional area"; however, it need not be, and can include part or all of more than one subsystem or functional area if all are the responsibility of the same agency. [8] The first level in the functional hierarchical structure. (See also Functional Area, CI, and CPCI, Type A - System Specification)

System Segment Specification - A specification similar in format to a system specification (Type A format), identifying a discrete package of system performance requirements, functional interfaces, and CIs contracted to one contractor or assigned to one Government organization directly responsible to the procuring activity for that part of a system's total performance. [5] (See System Segment, Type A - System Specification)

System Specification - A document which states all the necessary technical and mission requirements in terms of performance, allocates requirements to functional areas (or configuration items), defines the interfaces between or among the functional areas (or configuration items), and includes the test provisions to assure the achievement of all requirements. [7] (See also Type A - System Specification)

System Training Concept. A document summarizing ATC training policy based on review of user's requirements and planning factors as reflected in the SON and system operational concept and updates. Outlines conceptual guidance on T&E and deployment training planning efforts. It forms the basis for future training planning actions which are documented in the System Training Plan.

Survivability/Vulnerability (S/V) - Survivability is the capability of a system to accomplish its mission despite a man-made hostile environment. The USAF policy is that each system will have enough designed-in hardness and will be operated in a manner so that sufficient numbers will survive the expected threat.

There are direct nuclear and nonnuclear threats to virtually every Command, Control & Communications system, and there is a severe nuclear threat to the atmosphere and ionosphere, the propagation medium for radars and radio communications. Within the nuclear hardening area itself, there are several specialized disciplines. So although it is not difficult to understand the fundamentals of vulnerability and hardening, implementation of a sound survivability program usually requires a number of different specialists.

S/V is important in all phases of a system's life cycle, from concept through operations. Key milestones include the threat study, hardness specification, hardness verification (including testing), and hardness maintenance. The regulations do provide a formal mechanism for establishing survivability criteria, through the Nuclear Criteria Group and the Nonnuclear Survivability Technology Working Group. Mission Hardness design and verification must documented in such a way that AFLC and the operating command can readily maintain system hardness throughout its life, and evaluate the impacts of a changing threat.

Virtually every Command, Control and Communications system must be protected from the effects of electromagnetic pulse (EMP), a broad area nuclear effect. This can be done with sound state-of-the-art electrical engineering. Beyond EMP, hardening becomes very threat specific. [1]

Technical Data Control - This term refers to logging and managing the technical information which is developed by various engineering functions. (For other information concerning technical data control responsibilities, see AFR 310-1.) [6] (See also Engineering Management)

Technical Program Planning and Control - This term refers to the process of planning, monitoring, measuring, evaluating, directing, and replanning the management of the technical program. This process is carried out through such tasks as making risk analyses, developing and updating the work breakdown structure, accomplishing technical performance measurement, conducting technical reviews, performing change studies, and planning and implementing changes. [6] (See also Engineering Management)

Test. Any program or procedure which is designed to obtain, verify, or provide data for the evaluation of: research and development (other than laboratory experiments); progress in accomplishing development objectives; or performance and operational capability of systems, subsystems, components, and equipment items. [13]

Test Engineering - This function uses the technical data developed through the systems engineering process to develop test plans. These plans outline the test procedures and test requirements that are to be used to test the design solutions. (For other information concerning test planning, see AFR 80-14.) [6] (See also Engineering Management)

Test Requirements - The program office initiates the test planning process during the Conceptual Phase by preparing a Test and Evaluation Master Plan (TEMP). During the Validation Phase the contractor(s) initiate detailed test planning relative to hardware and computer program end-items (CIs and CPCIs). These test plans and procedures are submitted to the government for review and approval; the approved plans and procedures are the basis for subsystem and system testing. In order to test system requirements, a unique test must be associated with the appropriate end-item which incorporates requirement(s) to be tested. For those requirements which are inherent in a collection of end-items, the test of a requirement will be realized during system testing. Critical system requirements should be linked to unique end-items and be traceable to the original requirements as described in the MIL-STD-490 Type A and B specifications. Section 4 (MIL-STD-490/483 Type A and B Specifications, Quality Assurance Provisions) identifies the specific requirements for formal test and verification of the system (Type A) and subsequently its end-items (Type B). These test and verification requirements identify what specific system requirements (Section 3 of the specification) must be satisfied. Test requirements, therefore, identify the functional, performance, physical, operability, and design requirements which will be evaluated during system integration and test.

Test & Evaluation Master Plan (TEMP) - The TEMP is an overall plan which identifies and integrates the efforts and schedules of all test and check-out activities to be accomplished in the system development program. [7]

Traceability - (Requirements Traceability, Requirements Traceability Relationships) During the requirements engineering activities, sources of requirements (source documents) are referenced for each requirement identified. These source references provide the means of tracing the requirements from one set of system requirements documentation to the allocated requirements contained in the next level of system documentation, such as from a Type A to Type B specification. Sources for each requirement can also be maintained for pertinent studies, analyses, and plans: PMD, PMP, system sizing and timing studies, prototyping, simulations, test plans and procedures, and the like. The requirements and associated sources provide the means of verifying the requirements during the requirements engineering process and into later phases of the system acquisition by providing a repository of information on the system definition.

Software traceability refers to the capability to follow specific mission requirements through the various levels of specification to the actual code, and the capabilities to associate each area of code with a specified requirement. [2]

Trade-off Studies - Desirable and practical trade-offs among stated operational needs, engineering design, program schedule and budget, producibility, supportability, and life cycle costs, as appropriate, are continually identified and assessed. Trade-off studies are accomplished at the various levels of functional or system detail or as specifically designated to support the decision needs of the system engineering process. Trade-off studies, results and supporting rationale are documented in a form consistent with the impact of the study upon program and technical requirements. [10] (See also Systems Engineering)

Training Equipment - All types of maintenance and operator's training hardware, devices, visual/audio training aids and related software which (a) are used to train maintenance and operator personnel by depicting, simulating or portraying the operational or maintenance characteristics of an item, system or facility, and (b) must, by their nature, be kept consistent in design, construction and configuration with such items in order to provide required training capability.

Transportability - Any special requirements for transportability and materials handling shall be specified. The specifications shall include requirements for transportability which are common to all system equipment to permit employment, deployment, and logistic support. All system elements that, due to operational or functional characteristics, will be unsuitable for normal transportation methods, shall be identified. [3]

Two-part Specifications - Two-part specifications, which combine both development (performance) and product fabrication (detail design) specifications under a single specification number as procuring activity option. This practice requires both parts for a complete definition of both peformance requirements and detailed design requirements governing fabrication. Under this practice, the development specification remains alive during the life of the item as the complete statement of performance

requirements. Proposed design changes must be evaluated against both the product fabrication and the development parts of the specification. To emphasize the fact that two parts exist, both parts shall be identified by the same specification number and each part shall be further identified as Part I or Part II, as appropriate. [3]

Type A - System specification (also Segment Specification). This type of specification states the technical and mission requirements for a system as an entity, allocates requirements to functional areas, and defines the interfaces between or among the functional areas. Normally, the initial version of a system specification is based on parameters developed during the concept formulation period or an exploratory preliminary design period of feasibility studies and analyses. This specification (initial version) is used to establish the general nature of the system that is to be further defined during a contract definition, development, or contract design period. The system specification is maintained current during the contract definition, development, or equivalent period, culminating in a revision that forms the future performance base for the development and production of the prime items and subsystems (configuration items), the performance of such items being allocated from the system performance requirements (see MIL-STD-490, Appendix I for outline of form). [3] (See also System Specifications, System Segment Specification)

Type B - Development specifications. Development specifications state the requirements for the design or engineering development of a product during the development period. Each development specification shall be in sufficient detail to describe effectively the performance characteristics that each configuration item is to achieve when a developed item is to evolve into a detail design for production. The development specification should be maintained current during production when it is desired to retain a complete statement of performance requirements. Since the breakdown of a system into its elements involves items of various degrees of complexity which are subject to different engineering disciplines or specification content, it is desirable to classify development specifications by sub-types. [3] (See also Two-part Specifications, Development Specification and Specifications)

Type B5 - Computer program development specification. (See MIL-STD-490, Appendix VI for outline of form.) This type of specification is applicable to the development of computer programs, and shall describe in operational, functional, and mathematical language all of the requirements necessary to design and verify the required computer program in terms of performance criteria. The specification shall provide the logical, detailed descriptions of performance requirements of a computer program and the tests required to assure development of a computer program satisfactory for the intended use. [3] (See also Two-part specifications, Development Specifications, and Specifications)

Type C - Product specifications. Product specifications are applicable to any item below the system level, and may be oriented toward procurement of a product through specification of primarily function (performance) requirements or primarily fabrication (detailed design) requirements.

Sub-types of product specifications to cover equipments of various complexities or requiring different outlines of form are covered in MIL-STD-490, paragraphs 3.1.3.3.1 through 3.1.3.3.5 [3]

A product function specification states (1) the complete performance requirements of the product for the intended use, and (2) necessary interface and interchangeability characteristics. It covers form, fit, and function. Complete performance requirements include all essential functional requirements under service environmental conditions or under conditions simulating the service environment. Quality assurance provisions include one or more of the following inspections: qualification evaluation, preproduction, periodic production, and quality conformance.

A product fabrication specification will normally be prepared when both development and production of the item are procured. In those cases where a development specification (Type B) has been prepared, specific reference to the document containing the performance requirements for the item shall be made in the product fabrication specification. These specifications shall state: (1) a detailed description of the parts and assemblies of the product, usually by prescribing compliance with a set of drawings, and (2) those performance requirements and corresponding tests and inspections necessary to assure proper fabrication, adjustment, and assembly techniques. Tests normally are limited to acceptance tests in the shop environment. Selected performance requirements in the normal shop or test area environment and verifying test therefore may be included. Preproduction or periodic tests to be performed on a sampling basis and requiring service, or other, environment may reference the associated development specification. Product fabrication specifications may be prepared as Part II or a two-part specification (see Two-part Specifications, Product Specification and Specifications) when the procuring activity desires a close relationship between the performance and fabrication requirements. [3]

Type C5 - Computer program product specification. (See MIL-STD-490, Appendix XIII for outline of form.) A Type C5 specification is applicable to the production of computer programs and specifies their implementing media, i.e. punch tape, magnetic tape, disc, drum, etc. It does not cover the detailed requirements for material or manufacture of the implementing medium. When two-part specifications (See Two-part Specification) are used Type B5 shall form Part I and Type C5 shall form Part II. Specifications of this type shall provide a translation of the performance requirements into programming terminology and quality assurance procedures necessary to assure production of a satisfactory program. [3] (See also Product Specification and Specifications)

UPDATES - This relationship indicates that a function on the path updates internal system information as part of its activities. (See also Information Flow)

USES - This relationship indicates that a function on the path uses external information (external input) or internal system information

(internal input) in order to accomplish its activities. (See also Information Flow)

Using Command (Also called Using Agency and Using Activity) - The command primarily responsible for operational employment of a system. (See also Implementing Command and Supporting Command) [8]

UTILIZES - This relationship indicates that function on a path is dependent upon the use of one or more other functions in order to accomplish its activities. A single function or sequence of functions may be defined once and utilized as frequently as necessary in the control flow without having to be redefined (replicated) for each use. (See also Control Flow).

Validation - Comprises those evaluation, integration, and test activities carried out at the system level to ensure that the system being developed satisfies the requirements of the system specification. While the validation process has significant software implications, a software validation process, distinct from the system validation process, cannot be isolated since all evaluation and test activities that make up validation are focused at the system level. [7],[2]

Validation Phase - The period when major program characteristics are refined through extensive study and analyses, hardware development, test and evaluations. The objective is to validate the choice of alternatives and to provide the basis for determining whether or not to proceed into Full-Scale Development. (See AFR 800-2 and AFSCP 800-3) [1] (see also Acquisition Life Cycle)

Verification - The iterative process of determining whether the product of each step of the Computer Program Configuration Item (CPCI) development process fullfills all of the requirements levied by the previous step. [7],[2]

Work Breakdown Structure (WBS) - A work breakdown structure is a product-oriented family tree composed of hardware, software, services, and other work tasks which result from project engineering efforts during the development and production of a defense material item and which completely defines the project/program. A WBS displays and defines the product(s) to be developed or produced and relates the elements of work to be accomplished to each other and to the end product. (MIL-STD-881, MIL-STD-480) [1]

DEFINITION REFERENCES

The following references are the sources of many of the preceding definitions.

[1] ESD Program Manager's Handbook, Directorate, Acquisition Support, USAF Headquarters, Electronic Systems Division (AFSC), May 1976.

[2] G. Neil, and H. I. Gold, Software Acquisition Management Guidebook: Software Quality Assurance, ESD-TR-77-255 (DDC/NTIS Accession No. AD-A047318), USAF Electronic Systems Division, August 1977.

[3] MIL-STD-490, Specification Practices, 30 October 1968.

[4] MIL-STD-480, Configuration Control-Engineering Changes, Deviations and Waivers, 30 October 1968.

[5] MIL-STD-483, Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs, 31 December 1970.

[6] AFR 800-3, Engineering for Defense Systems, 1 June 1976.

[7] H. Bratman, and M. C. Finfer, Software Acquisition Management Guidebook: Verification. ESD-TR-77-263 (DDC/NTIS Accession No. AD-A048577), USAF Electronic Systems Division, June 1976.

[8] W. L. Schoeffel, An Air Force Guide to Software Documentation Requirements, ESD-TR-76-159 (DDC/NTIS Accession No. AD-A027051), USAF Electronic Systems Division, June 1976.

[9] Management Guide to Avionics Software Acquisition, Software Acquisition Process, Vol. II, ASD-TR-76-11, USAF Aeronautical Systems Division, June 1976.

[10] MIL-STD-499A (USAF), Engineering Management, 1 May 1974.

[11] J. B. Glore, Software Acquisition Management Guidebook: Life Cycle Events, ESD-TR-77-22 (DDC/NTIS Accession No. AD-A037115), USAF Electronic Systems Division, February 1977.

[12] D. H. Johnson and John J. Marciniak, "The Systems Operational Concept - A Computer Resources Viewpoint," presented at the National Aerospace and Electronics Conference (NAECON '78), Vol. 3, pp. 1316-1321, 18 May 1978.

[13] AFR 57-1 (draft), Statement of Operational Need, 8 June 1978.

LIST OF ABBREVIATIONS

Abbreviation	Definition
ADP	Automated Data Processing
AF	Air Force
AFR	Air Force Regulations
AFSC	Air Force Systems Command or Air Force Specialty Codes
AFSCM	Air Force Systems Command Manual
CADSAT	Computer-Aided Design and Specification Analysis Tool
CDRL	Contract Data Requirements List
C^3	Command, Control, and Communications
CI	Configuration Item
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPDP	Computer Program Development Plan
DCP	Decision Coordinating Paper
DID	Data Item Description
DoD	Department of Defense (also DOD)
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
DSARC	Defense Systems Acquisition Review Council
DT&E	Development Test and Evaluation
ECM	Electronic Countermeasures
ECCM	Electronic Counter-Countermeasures
ECP	Engineering Change Proposal
EMC	Electromagnetic Compatibility
EMP	Electromagnetic Pulse
ESD	Electronic Systems Division
EW	Electronic Warfare
FORTRAN	Formula Translation (an HOL)
FOT&E	Follow-on Operational Test and Evaluation
FQR	Formal Qualification Review
FQT	Formal Qualification Test
FSD	Full-Scale Development
GFP	Government-Furnished Property
HOL	Higher Order Language
HQ	Headquarters
I/O	System External and Internal Inputs and Outputs
IOT&E	Initial Operational Test and Evaluation
MIL-STD	Military Standard
MTBF	Mean-Time-Between-Failure
MTBM	Mean-Time-Between-Maintenance
MTTR	Mean-Time-To-Repair
O&M	Operations and Maintenance
OSD	Office of the Secretary of Defense
OT&E	Operational Test and Evaluation
PMD	Program Management Directive

LIST OF ABBREVIATIONS (cont'd)

Abbreviation	Definition
PMP	Program Management Plan
PO	Program Office (see also SPO)
PQT	Preliminary Qualification Test
PSL/PSA	Problem Statement Language/Problem Statement Analyzer
QA	Quality Assurance
RADC	Rome Air Development Center
R&D	Research and Development
RFP	Request for Proposal
ROC	Required Operational Capability
SEMP	System Engineering Management Plan
SE/TD	System Engineering/Technical Direction
SOC	Systems of Operational Concept
SON	System Operational Need
SOW	Statement of Work
SPO	System Program Office (see also PO)
SS	System Specification
S/V	Survivability/Vulnerability
TEMP	Test & Evaluation Master Plan
TR	Technical Report
USAF	United States Air Force
WBS	Work Breakdown Structure

APPENDIX C BIBLIOGRAPHY

A Users Guide to the Threads Management System, Computer Sciences Corp., November 1973.

Acquisition and Support Procedures for Computer Resources in Systems, AFR 800-14, Vol. II, 26 September 1975.

Acquisition of Major Defense Systems, DoDD 5000.1, DoD, 1 January 1977, pp. 8.

Air Force Quality Assurance Program, AFR 74-1, USAF, 11 September 1972.

Alford M.W., Browne I.F., Helton H.A., Hitt G.C., Software Requirements Engineering Methodology, SREM, Vol. I, Final Report, TRW-27332-6921-026-Vol-1, (DDC AD-A046 571/6WC), TRW Defense and Space Systems Group, Huntsville, AL, 1 August 1977, pp. 380.

Alford M.W., Burns I.F., Software Requirements Engineering Methodology Notebook, (DDC AD-923 470), October 1974, pp. 277.

Alford M.W., Burns I.F., "R-Nets: A Graph Model for Real-Time Software Requirements," Proceedings of the Symposium on Computer Software Engineering, Vol. XXIV, New York Polytechnic Press, Fox Jerome, ed., 20-22 April 1976, pp. 97-108.

Alford M.W., Berrie L.W., Pasini C.L., Berrie L.W., Requirements Development Using SCREM Technology, Vol. I, Contract No. DASG60-75-C-0022, CDRL C00H, TRW, Ballistic Missile Defense Advanced Technology Center, Huntsville, AL, 31 October 1977, pp. 677.

Alford M.W., Pasini C.L., Requirements Development Using SCREM Technology, Vol. II, Contract No. DASG60-75-C-0022, CDRL C00H, TRW, Ballistic Missile Defense Advanced Technology Center, Huntsville, AL, 31 October 1977, pp. 339.

Alford M.W., Berrie L.W., Pasini C.L., Management of Requirements Development Using SREM Technology, Contract No. DASG60-75-C-0022, CDRL C00G, TRW, Ballistic Missile Defense Advanced Technology Center, Huntsville, AL, 31 October 1977, pp. 183.

Alford Mack W., "A Requirements Engineering Methodology for Real-Time Processing Requirements," IEEE Trans. Software Eng., Vol. SE-3, No. 1, TRW-SS-76-07, January 1977, pp. 60-69.

An Introduction to SADT, (SofTech Document No. 9022-78), Softech, Inc., February 1976.

Automated Data Systems Documentation Standards, Standard 7935.1-S, Office of the Assistant Secretary of Defense, 13 September 1977.

Baker F.T., "Chief Programmer Team Management of Production Programming," IBM Syst. J., Vol. II, 1972, pp. 56-73.

Balkovich E., Engleberg G., "Research towards a technology to support the specification of data processing system performance requirements," in Proc. 2nd Int. Software Eng. Conf., San Francisco, CA, October 1976, pp. 110-115.

Balzer R., Goldman N. Wile D., "Informality in Program Specifications," IEEE Trans. Software Eng., March 1978, pp. 94-103.

Bandyopadhyay R., "Information for Organizational Decision Making- Literature Review," Vol. SMC-7, No. 1, January 1977.

Bastarache M.J. Problem Statement Analyzer (Version A2.1) Users Manual, Univ. Michigan, Ann Arbor, MI.

Bastarache M.J., Hershey E.A. III, Problem Statement Analyzer Command Descriptions (Version A2.1), ISDOS Working Paper No. 91, Univ. Michigan, Ann Arbor, MI, July 1975.

Becker H.B., "The Distributed Environment- A Formal Structure for Its Definition and Design," IEEE COMPCON Fall 76, Washington, D.C., pp. 364-366.

Belford P., Bond A.E., Henderson D.F., Sellers L.S., "Specifications- A Key to Effective Software Development," in Proc. 2nd Int. Software Eng. Conf., San Francisco, CA, 13-15 October 1976, pp. 71-79.

Belford P.C., Taylor D.S. "Specification Verification- A Key to Improving Software Reliability," Proceedings of the Symposium on Computer Software Engineering, Vol. XXIV, New York Polytechnic Press, Fox Jerome, ed., 20-22 April 1976, pp. 109-122.

Belford P.C., Donahoo J.D., Heard W.J., "An Evaluation of the Effectiveness of Software Engineering Techniques," IEEE COMPCON Fall 77, Washington, D.C., pp. 259-267.

Bell T.E., Bixler D.C., Dyer M.E., "An Extendable Approach to Computer-Aided Software Requirements Engineering," IEEE Trans. Software Eng., Vol. SE-3, No. 1, TRW-SS-76-07, January 1977, pp. 49-60.

Bell T.E., Thayer T., "Software Requirements: Are They Really a Problem?," in Proc. 2nd Int. Software Eng. Conf., TRW-SS-76-04, San Francisco, CA, October 1976, pp. 61-68.

Bell Thomas E., Bixler David C., "A Flow Oriented Requirements Statement Language," TRW Software Series, Proc of the Symposium on Computer Software Engineering, Vol. XXIV, TRW-SS-76-02, New York Polytechnic Press, Fox Jerome, ed., April 1976, pp. 109-122.

Benoit W.E., et al, REVS Maintenance Manual, Vol. III, SREP Final Report, Contract No. DASG60-75-C-0022, TRW. Ballistic Missile Defense Advanced Technology Center, Huntsville. AL, 1 August 1977, pp. 422.

Benson J., Melton R., "A Laboratory for the Development and Evaluation of BMD Software Quality Enhancement Techniques," in Proc. 2nd Int. Software Eng. Conf., October 1976.

Black W.W., "The Role of Software in Successful Computer Applications," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976, pp. 201-205.

Boehm B.W., "A Concept of Model Driven Software," in Proc. TRW Symp. Reliable, Cost Effective, Secure Software, Los Angeles, CA, March 1974.

Boehm B.W., "Software Engineering," IEEE Transactions on Computers, Vol. C-25, No. 12, December 1976, pp. 1226-1241.

Boehm B.W., "Software Engineering: R&D Trends and Defense Needs," Proceedings of the Conference on Research Directions in Software Technology, 10-12 October 1977, pp. 1-1 - 1-43.

Boehm B.W., "Some Steps Toward Formal and Automated Aids to Software Requirements and Design," IFIP Proceedings, TRW Systems Group, Redondo Beach, CA, North-Holland Publishing Co., 1974, pp. 192-197.

Boehm B.W., "The High Cost of Software," Practical Strategies for Developing Large Software Systems, Addison-Wesley Publishing Co., Horowitz Ellis, ed., 1975, pp. 3-14.

Boehm B.W., Brown J.R., Lipow M., "Quantitative Evaluation of Software Quality," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976, pp. 592-605.

Boehm B.W., "Software and Its Impact: A Quantitative Assessment," Datamation, Vol. 19, No. 9, May 1973, pp. 48-59.

Bratman H., Court T., "The Software Factory," Computer, Vol. 8, No. 5, May 1975, pp. 28-37.

Brooks F.P., The Mythical Man-Month: Essays on Software Engineering, Reading, MA, 1974, pp. 195.

Brooks F.P.. Jr.. The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley Publishing Co.. 1975.

Brooks F.P., Jr., "The Mythical Man-Month." Datamation, December 1974, pp. 45-53.

Burger R.T.. "AUTASIM: a system for computerized assembly of simulation models," SIGPLAN Notices. presented at the Winter Simulation Conf. General Research Corporation. January 1974.

Burns F., et al.. "Current Software Requirements Engineering Technology," TRW Systems Group, Huntsville, AL, August 1974.

Business Systems Planning: Information Systems Planning Guide. IBM Application Manual, First Edition, GE 20-0527-1, August 1975, pp. 92.

CADSAT Application Guidebook. ESD-TR-78-125. (DDC AD-A061 00977WC), ISDOS Project, University of Michigan, Hanscom AFB. MA. August 1978, pp. 94.

Caine Stephen H. Gorden E. Kent, PDL - A Tool for Software Design," Montvale, NJ. AFIPS Press, 1975.

Carlson W.E.. "Software Research in the DOD," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976, pp. 378-383.

Cheng Lorna L.. Software Engineering Techniques and Tools: Language and Life Cycle Considerations Working Paper, WP-21017, Project No. 522p. Contract No. F19628-77-C-0001, USAF(ESD), Bedford, MA.

Clappier J.. "Embedded Computers for Military Applications," IEEE COMPCON Fall 77, Washington, D.C.. pp. 238-240.

"COMSYNCSAT System Study," Systems Engineering Management Workshop, Vol. I and II. General Electric, Philadelphia, PA, The Space Division and the Re-Entry & Environmental Systems Division. c. 15 July 1969.

Configuration Control, Engineering Changes, Deviations and Waivers. MIL-STD-480, 30 October 1968.

Configuration Management During Definition and Acquisition Phases, AFSCM 375-1 and revisions 375-1A, 1 June 1964.

Configuration Management Implementation Guidance, Department of Defense Instruction 5010.21, 6 August 1968 and changes.

Configuration Management Practices for Systems, Equipment, Munitions and Computer Programs MIL-STD-483 (USAF), 31 December 1970.

Configuration Management, Department of Defense Directive 5010.19, 17 July 1968, and changes.

Connolly J.T.. Software Acquisition Management Guidebook: Regulations, Specifications and Standards, ESD-TR-75-91, (DDC AD-A016401), Contract No. F19628-75-C-0001, The Mitre Corporation, USAF/AFSC: Electronics System Division (TOI), Hanscom AFB, MA, October 1975, pp. 121.

Cooper D.W., "Adaptive Testing," in Proc. 2nd Int. Software Eng. Conf., October 1976.

Couger J.D., "Evolution of Business Systems Analysis Techniques," ACM Computing Surveys, Vol. 5, No. 3, September 1973, pp. 167-198.

Couger, J.D. Knapp R.W., System Analysis Techniques, John Wiley & Sons, Inc., 1974, pp. 509.

Covelli Robert R., Williams Thomas G., "Management Overlay for the BMDATC Software Development System," TM-HU-235/000/00, CDRL A00C, Systems Development Corporation, Ballistic Missile Defense Advanced Technology Center (BMDATC), Advanced Research Center (ARC), Huntsville, AL, 15 December 1976, pp. 195.

Daly E.B., "Management of Software Development," IEEE Trans. Software Eng. Vol. SE-3, No. 1, May 1977, pp. 229-242.

"Data Processing System Requirements," Final Rep., DASG60-75-C-0124, Aeronutronic-Ford Corp., Willow Grove, PA, June 1976.

Davis Carl G., Vick Charles R., "The Software Development System," IEEE Trans. Software Eng., Vol. SE-3, No. 1, January 1977, pp. 69-84.

DeRoze B.C., Defense System Software Management Plan, (DDC AD-A-022558/IWC), March 1976, pp. 97.

DeWolf J.B., Wexler J., "Approaches to Software Verification with Emphasis on Real-Time Applications," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 41-51.

"DOD weapons systems software acquisition and management study," Vol. I, The MITRE Corp., May 1975.

"DOD weapons systems software management study," The Johns Hopkins Univ., Applied Physics Lab. (APL), May 1975.

"DPSPR final report," Contract DASG60-75-C-0123, General Research Corp., Santa Barbara, CA, April 1976.

Driscoll A.J.. Software Visibility for the Program Manager, (DDC AD-A035 164), November 1976, pp. 57.

Drutz A., "427M: A Contractor's View of Lessons Learned," National Aerospace & Electronics Conference. NAECON 78, Vol. 3 of 3, 78CH1336-7, System Development Corporation, Dayton, OH. 18 May 1978, pp. 1327-1332.

Dyer M.E.. Gunther L.J.. Smith R.W., Benoit W.E.. Bergstresser P.N.. REVS Users Manual. SREP, Vol. II, Final Report. TRW-27332-6921-026-Vol-2. (DDC AD-A046 572/4WC), TRW Defense and Space Systems Group. Huntsville. AL. 1 August 1977, pp. 418.

Embedded Computer Resources and the DSARC Process- A Guidebook, Department of Defense. 1977.

Endres A.B., "An Analysis of Error and their Causes in System Programs," IEEE Trans. Software Eng.. June 1975, pp. 140-149.

Enger N.L.. Management Standards for Developing Information Systems, AMACOM. 1976, pp. 227.

Engineering For Defense Systems. AFR 800-3. 1 June 1976.

Engineering Management. MIL-STD-499A (USAF). 1 May 1974.

ESD Program Manager's Handbook. USAF/AFSC: Electronic Systems Division (TOS), Hanscom AFB, MA, May 1976, pp. 129.

Fagan M.E.. "Inspecting Software Design and Code," Datamation, Vol. 23, No. 10, October 1977. pp. 133-144.

Fife D.W., Computer Software Management: A Primer for Project Management and Quality Control, FIPS Special PUB 500-11.

Finfer M.F.. Mish R.K., Software Acquisition Management Guidebook: Software Cost Estimation and Measurement, (DDC AD-A055-574), Contract No. F19628-76-C-0236, System Development Corporation, USAF/AFSC. Electronic Systems Division (TOI), Hanscom AFB, MA, March 1978, pp. 114.

Fleischer R.. Spitler R.. "A Project Management System for Software Development," IEEE COMPCON Fall 77, pp. 110-114.

FLON. : Program Design with Abstract Data Types. (DDC AD-A01843), June 1975.

Functional Analysis of Information Networks: A Structured Approach to the Data Communications Environment, John Wiley & Sons, Inc.. 1973, pp. 281.

Galvan R.J.. Tucker A.G. Fujii R.U.. "Flow Language Study Final Report, R: CDB-76084, Contract No. F04704-76-C-0001, CDRL A018, SAMSO/MNNC. Norton AFB, CA, 1 October 1976.

Gansler J.S., "Keynote; Software Management," Proceedings of the Symposium on Computer Software Engineering, Vol. XXIV, New York Polytechnic Press, Fox Jerome, ed., 20-22 April 1976, pp. 1-10.

Gaulding S.N., Lawson J.D., "Process design engineering- A methodology for real-time software requirements," in Proc. 2nd Int. Software Eng. Conf., October 1976.

Gaulding S.N., "A Software Design Methodology and Tools," IEEE COMPCON Spring 77, San Francisco, CA, pp. 198-200.

"Getting the Requirements Right," EDP Analyzer, Vol. 15, No. 7, July 1977.

Ghett R.V., Dotti F.A., Tulic M., "Structured Analysis Guidelines and Standards," Chase Manhattan Bank, N.A., New York, NY, May 1978. pp. 60.

Gibson H.L., "Determining User Involvement," Journal of Systems Management, Vol. 28, No. 8, August 1977, pp. 20-22.

Glaser E.L. et al., "The LOGOS Project," IEEE COMPCON 72, pp. 175-192.

Glore J.B., Software Acquisition Management Guidebook: Life Cycle Events, ESD-TR-77-22, (DDC AD-A037115). Contract No. F19628-77-C-0001, The MITRE Corporation, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 1977, pp. 68.

Graham, R.M., Clancy G.J., Devaney D.B., "A Software Design and Evaluation System," C.ACM, February 1973, pp. 110-116.

Groobey J., "Maximizing Returns on EDP Investments." Data Management, September 1972.

Guidelines for Documentation of Computer Programs and Automated Data Systems, FIPS PUB 38.

Hagan S.R., Knight C.W., An Air Force Guide for Monitoring and Reporting Software Development Status, ESD-TR-75-85, (DDC AD-A016-488), Contract No. F19628-77-C-0001. The Mitre Corporation, USAF/AFSC. Electronic Systems Division (TOI), Hanscom AFB, MA, September 1975, pp. 98.

Hales, K.A., "Software Management Lessons Learned- The Hard Way." A Collection of Technical Papers. AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Boeing Aerospace Company, Seattle, WA, Los Angeles, CA, 31 October - 2 November 1977, pp. 182-188.

Hamilton M., Zeldin S., "Higher Order Software- A Methodology for Defining Software," IEEE Trans. Software Eng., Vol. SE-2, No. 1, March 1976, pp. 9-32.

Hamilton M., Zeldin S., "The foundations for axes: a specification language based upon completeness of control," R-964, The Charles Stark Draper Laboratory Inc., March 1976.

Hamilton M., Zelden S., "The Manager as an Abstract Systems Engineer," IEEE COMPCON Fall 77, Washington, D.C., pp. 98-105.

Hamilton M., Zelden S., "Verification of an Axiomatic Requirements Specification," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 354-371.

Hantler, "An Introduction to Proving the Correctness of Programs," Computer Surveys, Vol. 8, No. 3, April 1976.

Hartman W., Matthes H., Proeme A., Management Information System Handbook: "ARDI", McGraw-Hill Book Company, 1968.

Head R.V., "Automated System Analysis," Datamation, 15 August 1971, pp. 22-24.

Hershey E.A., III, Winters E.W., Berg D.L.R., Dickey A.F., Kahn B.L., Problem Statement Language Version 3.0 Language Reference Manual, ISDOS Working Paper No. 68, Univ. Michigan, Ann Arbor, MI, Revised May 1975.

Hetzel W.C., "The Future of Quality Software," IEEE COMPCON Spring '77, San Francisco, CA, pp. 211-212.

Hilbing, F.J., "Software Engineering for Microprocessors," IEEE COMPCON Fall '77, pp. 180-184.

HIPO - A Design Aid and Documentation Technique, Second Edition, IBM GC20-1851, May 1975, pp. 130.

HIPO: Design Aid and Documentation Tool, SR20-9413-0, IBM, 1973.

Hodges B.C., Ryan J.P., "A System for Automatic Software Evaluation," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976, pp. 617-623.

Howden, "Symbolic Testing and the Diggest Symbolic Evaluation System," IEEE Trans. Software Eng., Vol. SE-3, No. 4, July 1977.

Howley P.P., Scholten R.W., "Test Tool Implementation," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Boeing Aerospace Co. Seattle, Washington, Los Angeles, CA, 31 October - 2 November 1977, pp. 372-377.

"Introduction: Xerox Systems Methodology," Report No. 6748/C312/3100/01, Corporate Systems Methodologies, Xerox Square -052, c. May 1978, pp. 66.

Irvine C.A., Brackett John W., "Automated Software Engineering Through Structured Data Management," IEEE Trans. Software Eng., Vol. SE-3, No. 1, January 1977, pp. 34-40.

Jackson M., Principles of Program Design, New York, NY, Academic Press, 1975.

Jensen K., Wirth N., PASCAL Users Manual and Report, New York, Springer-Verlag, 1974.

Jensen L.K., "On the Education of Software Development Managers," IEEE COMPCON Fall '77, Washington D.C., pp. 12-16.

Johnson David H. Marciniak John J., "The Systems Operational Concept - A Computer Resource Viewpoint," National Aerospace & Electronics Conference, NAECON 78, Vol. 3 of 3, 78CH1336-7, Dayton, OH, 18 May 1978, pp. 1316-1321.

Johnson L., Coker F., Smith D., Management Information System Requirements for ESD Program Offices, Logicon Report No. DSJ-R77024, ESD-TR-78-132, (DDC AD-A056103), Contract No. F19628-77-C-0178, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 1978, pp. 176.

Johnson Larry A., Merrithew Paul B., "Automated Support for Requirements Analysis and Traceability," National Aerospace & Electronics Conference, NAECON 78, Logicon, Inc., Lexington MA, 18 May 1978, pp. 15.

Keider S.D., "Why Projects Fail," Datamation, December 1974.

Koppang R., "Process design system- An integrated set of software development tools," in Proc. 2nd Int. Software Eng. Conf., October 1976.

Kosslakoff A., Sleight T.P., "Software Requirements Analysis and Validation," Abridged Proceedings from the Software Management Conference, First Series, The John Hopkins University, 1976.

Langefors B., "Information system design computations using generalized matrix algebra," BIT, Vol. 5, 1965, pp. 96-121.

Lawson J., "Baseline software process productivity and error analysis," Texas Instruments, Huntsville, AL, April 1976.

Liskov B.H., Berzins V., "An Appraisal of Program Specifications," Proceedings of the Conference on Research Directions in Software Technology, 10-12 October 1977, pp. 13-1 - 13-34.

Liskov B.H., Zillies S.N., "Specification Techniques for Data Abstractions," IEEE Trans. Software Eng. Vol. SE-1, No. 1, March 1975.

"Logicflow," Logicon Rep., 9 November 1977.

Management Guide To Avionics Software Acquisition: An Overview of Software Development and Management, Vol I, Final Report, ASD-TR-76-11, Contract No. F33657-76-C-0234, Logicon, USAF/AFSC: Aeronautical Systems Division, Wright-Patterson AFB OH, June 1976, pp. 53.

Management Guide To Avionics Software Acquisition: Software Acquisition Process, Vol II, Final Report, ASD-TR-76-11, Contract No. F33657-76-C-0234, Logicon, USAF/AFSC: Aeronautical Systems Division, Wright-Patterson AFB, OH, June 1976, pp. 116.

Management Guide To Avionics Software Acquisition: Summary of Software Related Standards and Regulations, Vol III, Final Report, ASD-TR-76-11, Contract No. F33657-76-C-0234, Logicon, USAF/AFSC: Aeronautical Systems Division, Wright-Patterson AFB, OH, June 1976, pp. 56.

Management Guide To Avionics Software Acquisition: Technical Aspects Relative to Software Acquisition, Vol IV, Final Report, ASD-TR-76-11, Contract No. F33657-76-C-0234, Logicon, USAF/AFSC: Aeronautical Systems Division, Wright-Patterson AFB, OH, June 1976, pp. 165.

Management of Automatic Data Processing Systems Air Force Regulation 300-2, 12 November 1971.

Management of Computer Resources in Major Defense Systems Department of Defense Directive 5000.29, 26 April 1976.

Management of Computer Resources in Major Defense Systems. DoDD 5000.29, DoD, 26 April 1976.

Manley J.H., "Embedded Computers - Software Cost Considerations," AFIPS NCC Proceedings, Vol. 43, HQ AFSC, Andrews AFB, 1974, pp. 343-347.

Manley J.H., "Preview: Blueprint for Software Success," IEEE COMPCON Fall '76, Washington D.C., pp. 38.

Martin W.A., Bosyj M., "Requirements Derivation in Automatic Programming," Proceedings of the Symposium on Computer Software Engineering, Vol. XXIV, New York Polytechnic Press, Fox Jerome ed., 20-22 April 1976, pp. 123-131.

Mayfield J.P., "Software Development Methodology Selection Criteria," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 Novemeber 1977, pp. 93-96.

McCarthy R., "Applying the Technique of Configuration Management to Software," Defense Management Journal, Vol. II, No. 4, October 1975, pp. 23-28.

McGowan C., Kelly J.. "A Review of Some Design Methodologies," InfoTech State of the Art Conference on Structured Design, Holland, 1976.

McGowen C.L.. McHenry R.C., "Software Management," Proceedings of the Conference on Research Directions in Software Technology, 10-12 October 1977, pp. 6-1 - 6-67.

Merten A., Teichroew D., "The Impact of Problem Statement Languages on Evaluating and Improving Software Performance," AFIPS FJCC, 1972, pp. 849-957.

Military Standard Specifications Practices, MIL-STD-490, Department of Defense, October 1968.

Miller, G.A., "The magical number seven, plus or minus two: Some limits on our capacity for processing information," Psychol. Rev.. Vol. 63, Mar 1956, pp 81-97.

Mullin F.J.. Considerations for a Successful Software Test Program," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference. TRW Defense and Space Systems Group, Los Angeles, CA, 31 October - 2 November 1977, pp 68-74.

Myers G., Reliable Software Through Composite Design, New York, NY, Petrocelli/Charter, 1975.

Myers G.J., Software Reliability: Principles and Practices, Wiley-Interscience. 1976.

Myers Ware. "The Need for Software Engineering," Computer, Vol. 11, No. 2, February 1978, pp. 12-26.

Neil G., Gold H.I.. Software Acquisition Management Guidebook: Software Quality Assurance. ESD-TR-77-255, (DDC AD-A047318), Contract No. F19628-76-C-0236, System Development Corporation, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, August 1977. pp. 95.

Nelson E.. "Developing a Software Methodology," IEEE COMPCON Fall '77. pp. 144-145.

Nunamaker J.F.. "A methodology for the design and optimization of information processing systems." in 1971 Fall Joint Comput. Conf.. AFIPS Conf. Proc., Vol. 28. Montvale, NJ, AFIPS Press, 1971.

Oberndorf R.T., A Report on the Use of URL/URA as a quality assurance and Data-Base Definition Tool, Preliminary, Computer Sciences Corporation, Naval Ocean Systems Center, Code 822, San Diego, CA, 16 January 1978.

Operational Software Management and Development for U.S. Air Force Computer Systems: A Report by the Committee on Operational

Software Management and Development of the Air Force Studies Board Assembly of Engineering National Research Council, National Academy of Sciences, Washington D.C., 1977.

Palmer D.F., et al., "SETS 1 software design specifications," CR-1-516, General Research Corp., July 1972.

Parker R.W., "The Development of Software Requirements: Experiences from a Large Scale Program," IEEE COMPCON Fall '76, Washington D.C., pp. 45-49.

Parnas D.L., "A Technique for Software Module Specification with Examples," C.ACM, Vol. 15, No. 15, May 1972, pp. 330-336.

Parnas D.L., "On the Criterion to be Used in Decomposing Systems into Modules," C.ACM. Vol. 15, No. 12, pp. 1053-1058.

Peters L., Tripp L., "Is software design wicked?," Datamation, May 1976, pp. 127-129.

Peterson J.B., "Software Engineering Methods, Mini-Course Synopsis," National Aerospace & Electronics Conference, NAECON 78, Vol. 1 of 3, Air Force Institute of Technology, Department of Electrical Engineering, Wright-Patterson AFB, OH, 16-18 May 1978, pp. 314-315.

Practical Strategies for Developing Large Software Systems, Addison-Wesley Publishing Co., Horowitz E., ed., 1975.

Premo A.F., Jr., "Computer Softtware: Estimating Guidelines," IEEE COMPCON Fall '76, pp. 146-150.

Problem Statement Language (PSL) User's Manual, Univ. Michigan, March 1975.

Proc. of the IEEE 1978 National Aerospace and Electronics Conference, NAECON 78. Vol. 1 of 3. 78CH1336-7 NAECON, The Institute of Electrical and Electronics Engineers, Dayton Section Aerospace and Electronics Systems Society, Dayton, OH. 16-18 May 1978.

"Process design methodology," Process Design Language Specifications, Vol. 1, Texas Instruments, Huntsville, AL, December 1975.

Program Management, AFR 800-2, 16 March 1972.

PSL/PSA Training Seminar Student Manual, Univ. Michigan, Ann Arbor, MI, December 1976.

Putnam L.H., "A Macro-Estimating Methodology for Software Development," IEEE COMPCON Fall '76, pp. 138-143.

Quality Assurance for Electronic Systems, ESDM 74-1, USAF(AFSC/ESD), 25 February 1974.

Randell, "System Structure for Software Fault Tolerance," IEEE Trans. Software Eng., Vol. SE-1, No. 2, June 1975.

"RDL/RDP User's Manual," Update D-4, Doc. No. RSS-060, Sperry Rand Corp., 15 July 1978, pp. 200.

Reifer D., "Software Specifications: A Tutorial," in Proc. COMPCOM 76, September 1976.

Reifer D.J., "The Software Engineering Checklist," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 97-101.

Reifer D.J., Trattner S., "Software Specification Techniques: A Tutorial," IEEE COMPCON Fall '76, Washington, D.C., pp. 39-44.

Required Operational Capabilities, AFR 57-1, 30 May 1975.

"Requirements Engineering and Validation System Users Manual," draft, TRW Report No. 27332-6921-022, TRW, 15 July 1976.

Requirements for Digital Computer Program Documentation, Revision 1, WS-8506, Naval Ordanance System Command, 1 November 1971.

Requirements for Digital Computer Program Documentation, WS-8506, Revision 1, Naval Ordinance Systems Command, 1 November 1971.

Richter M.D., Mason J.D., et al., Alford M.W., Burns I.F., Helton H.A., Lawson J.T., "Software Requirements Engineering Methodology," CDRL C011, (DDC AD-B013-480), DGSG60-75-C-0022, TRW Defense and Space Systems Group, Ballistic Missile Defense Advanced Technology Center (BMDATC), Huntsville, AL, 1 September 1976, pp. 1-321.

Rigo J.T., "How to Prepare Functional Specifications," Datamation, May 1974, pp. 78-80.

Rose C.W., "LOGOS and the software engineer," in 1972 Fall Joint Comput. Conf., AFIPS Conf. Proc., Vol. 41, AMSEL-PP-Cn122(CAE), Case Western Reserve University, Cleveland, OH, Montvale, NJ, AFIPS Press, 1972, pp. 311-323.

Ross D., Shoman K., "Structured analysis for requirements definition," in Proc. 2nd Int. Software Eng. Conf., October 1976.

Ross D.T., "Quality Starts with Requirements Definition," Panel Position Paper for IFIP TC2 Working Conference on Constructing Quality Software, 9037-16, Novosibirsk, USSR, February 1977.

Ross D.T., Goodenough J.B., Irvine C.A., "Software Engineering Process, Principles, and Goals," Computer, Vol. 8, No. 5, Addison-Wesley Publishing Co., pp. 17-27.

Ross Douglas T., Schoman Kenneth E., Jr., "Structured Analysis for Requirements Definition," IEEE Trans. Software Eng., Vol. SE-3, No. 1, January 1977, pp. 6-15.

Ross Douglas T., "Structured Analysis (SA): A Language for Communicating Ideas," IEEE Trans. Software Eng., Vol. SE-3, No. 1, January 1977, pp. 16-34.

Ross, D.T., "It's time to ask why?," Software Practice Experience, Vol. 1, Jan.-Mar. 1971, pp. 103-104.

Ross, D.T., Goodenough, J.B., Irvine, C.A., "Software engineering: Process, principles, and goals," Computer, May 1975, pp. 17-27.

Ross, D.T., "A generalized technique for symbol manipulation and numerical calculation," Commun. Ass. Comput. Mach., Vol. 4, March 1961, pp. 147-150.

Royce W.W., "Software Requirements Analysis: Sizing and Costing," Practical Strategies for Developing Large Software Systems, Addison-Wesley Publishing Co., Horowitz Ellis, ed., 1975, pp. 57-71.

Rubin M.L., "Introduction to the System Life Cycle," Handbook of Data Processing Management, Vol. 1, Princeton, NJ, Brandon Systems Press, 1970.

"SADT(TM) The SofTech Approach to System Development," January 1976.

Salter K., "A Methodology for Decomposing System Requirements into Data Processing Requirements," in Proc. 2nd Int. Software Eng. Conf., San Francisco, CA, 13-15 October 1976, pp. 91-101.

Salwin A.E., A Test Case Comparison of URL/URA and REVS, FS-77-161, Navy Contract No. N00017-72-C-4401 Fleet Systems Department, The John Hopsins University, Applied Physics Laboratory, pp. 139, Laurel, MD, July 1977.

Schneidewind, N.F., "An Approach to Software Reliability Prediction and Quality Control," AFIPS FJCC Proceedings, Naval Postgraduate School, Monterey, CA, 1972, pp. 837-847.

Schoeffel W.L., An Air Force Guide to Software Documentation Requirements, ESD-TR-76-159, (DDC AD-A027051), Contract No. F19628-76-C-0001, The MITRE Corporation, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, June 1976, pp. 174.

Schwartz J.I., "Construction of Software: Problems and Practicalities," Practical Strategies for Developing Large Software Systems, Addison-Wesley Publishing Co., Horowitz Ellis, ed., 1975, pp. 15-53.

Searle L.V., An Air Force Guide to the Computer Program Development Specification, ESD-TR-78-139, (DDC AD-A055573), Contract No. F19628-76-C-0236, System Development Corporation, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 1978, pp. 103.

Searle L.V., An Air Force Guide to Computer Program Configuration Management, ESD-TR-77-254, (DDC AD-A047-308), Contract No. F19628-76-C-0236, System Development Corporation, USAF/AFSC, Electronic Systems Division (TOI), Hanscom AFB, MA, August 1977, pp. 139.

Slaughter J.B., "Understanding the Software Problem," AFIPS NCC Proceedings, Vol. 43, Naval Electronics Laboratory Center, San Diego, CA, 1973, pp. 333-336.

Slomski Joseph F., "Systems Engineering Management, Its Principles and its Practice," Aerospace Management, Vol. 4, no. 2, General Electric Company, Aerospace Group, Space Division's Management Practices Section, of the Finance and Contract Management Operation, Valley Forge Space Technology Center, Philadelphia, PA, 1969.

Smith C.P., "Resolving User/System Differences," Journal of Systems Management, Vol. 28, No. 7, July 1977, pp. 16-21.

Smith Daniel G., Johnson Larry A., Merrithew Paul B., "Automated Analysis of System Specifications," The Second U.S. Army Software Symposium, Logicon, Inc., Williamsburg, VA, 26-27 October 1978, pp. 23.

Software Acquisition Managment Guidebook, Regulations, Specifications, and Standards, ESD TR 75-91, October 1975.

Software Engineering Guidebooks, PRC Information Sciences Company, February 1977.

Software Management Conference, abridged proceedings, first series, 1976.

Software Quality Assurance Program Requirements, 5 April 1974.

"Software Requirements Engineering Methodology," final report, TRW Report No. 27332-6921-019, 12 December 1975.

Specification Practices, MIL-STD-490, 30 October 1968.

Specifications, Types and Forms, MIL-S-83490, 30 October 1968.

Statement of Operational Need, AFR 57-1 (draft), 8 June 1978.

Steele S.A., "Characteristics of Managing Real-Time Software Development for Military Systems," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Software Design and Development, RCA Missile and Surface Radar, Moorestown, NJ, Los Angeles, CA, 31 October - 2 November 1977, pp. 174-181.

Structure Analysis Reader Guide, (SofTech Document No. 9022-73.1). Softech, Inc., Waltham, MA, 1975.

Sylvester R.J., "Elements of the Computer Program Development Plan," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 18-22.

"System decomposition technology," Final Report, Computer Science Corp., Huntsville, AL, April 1976.

Systems Engineering Management Procedures, AFSCM 375-5, 10 March 1966.

Tactical Digital Systems Documentation Standards, SECNAVINST 3560.1, Department of the Navy, 8 August 1974.

Taggart W.M., Jr., Tharp M.O., "A Survey of Information Requirements Analysis Technique," ACM Computing Surveys, Vol. 9, No. 4, December 1977, pp. 273-290.

Taylor J.M., Software Requirements for Dedicated High Integrity Systems, N75-17983/SWC, March 1974, pp. 34.

Technical Reviews and Audits for Systems and Computer Programs, MIL-STD-1521 (USAF), 1 September 1972.

Teichroew D., "A Survey of Languages for Stating Requirements for Computer-Based Information Systems," in 1972 Fall Joint Comput. Conf., AFIPS Conf. Proc., Vol. 41, Montvale, NJ, AFIPS Press, 1972, pp. 1203-1224.

Teichroew D., Hershey E., Bastarache M., "An introduction to PSL/PSA," ISDOS Working Paper 86, Univ. Michigan, Dept. Industrial and Operations Eng., Ann Arbor, March 1974.

Teichroew D., "ISDOS and Recent Extensions," Proceedings of the Symposium on Computer Software Engineering, Vol XXIV, New York Polytechnic Press, Fox Jerome, ed., 20-22 April 1976, pp. 75-81.

Teichroew D., PSL/PSA Example, ISDOS Working Paper No. 74, November 1976, pp. 208.

Teichroew D., Bastarache M., Problem Statement Analyzer Reports (Version A2.1), ISDOS Working Paper No. 99, Univ. Michigan, Ann

Arbor, MI, November 1975.

Teichroew D., Gackowski Z., "Structured Systems Design," Ideas for Management, Paper and Case Histories presented at the 1977 Annual Conference of the Association for Systems Management, 1977, pp. 45-58.

Teichroew D., Hershey E.A., III, "PSL/PSA -- A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976.

Teichroew D., Sayani H., "Automation of System Building," Datamation, 15 August 1971, pp. 25-30.

Teichroew Daniel, Hersey Ernest A., III, "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems," IEEE Trans. Software Eng., Vol. SE-3, No. 1, January 1977, pp. 41-48.

Thayer R.H., Lehman J.H., "Software Engineering Project Management: A State-of-the-Art Report," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 153-167.

Thayer T., "Understanding Software Through Analysis of Empirical Data," TRW Software Series, TRW-SS-75-04, May 1975.

"The BMDATC software development system," Vols. I and II, and supplement, BMDATC, July 1976.

The Decision Coordinating Paper (DCP) and the Defense Systems Acquisition Review Council (DSARC), DoDD 5000.2, Department of Defense, 1 January 1977.

"The Development of a Software Theory to Classify and Detect Software Errors," Rep. HR-74012, LOGICON, March 1974.

The Time Automated Grid System (TAG): Sales and Systems Guide, end edition, publ. GY20-0358-1, IBM, May 1971.

Thurber K.J., "Requirements-Oriented Design: An Emerging Design Strategy," IEEE COMPCON Spring '77, San Francisco, CA, pp. 305-308.

Turn, R.M., Davis M.R., Reinstedt R.N., "A Management Approach to the Development of Computer-Based Systems," A Collection of Technical Papers, AIAA/NASA/IEEE/ACM Computers in Aerospace Conference, Los Angeles, CA, 31 October - 2 November 1977, pp. 11-17.

User Requirements Analyzer (URA), User's Manual, H6180/Multics/Version 3.3, ESD-TR-78-131, Contract No. F19628-76-C-0197, CDRL 020, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI). USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, July 1978, August 1978, pp. 580.

User Requirements Analyzer (URA), User's Manual H6180/Multics/Version 3.2 & Figures, ESD-TR-78-128, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 77, pp. 358.

User Requirements Language (URL), User's Manual Part II H6180/Multics/Version 3.3, ESD-TR-78-130, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, July 1978, pp. 447.

User Requirements Language (URL) User's Manual Part I, H6180/Multics/Version 3.2, ESD-TR-78-127, (DDC AD-A054096), ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 77, pp. 191.

User Requirements Language (URL), User's Manual Part I IBM/370/MVS/ TSO Version 3.2, ESD-TR-78-129, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, July 1978, pp. 198.

User Requirements Language (URL), User's Manual Part II IBM/370/MVS/ TSO Version 3.2, ESD-TR-78-129, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, July 1978, pp. 439.

User Requirements Language (URL), User's Manual Part I H6180/Multics/Version 3.3, ESD-TR-78-130, ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, July 1978, pp. 212.

Users Requirements Language (URL), User's Manual Part II, H6180/Multics/Version 3.2, ESD-TR-78-172, (DDC AD-A055505), ISDOS Project, University of Michigan, USAF/AFSC: Electronic Systems Division (TOI), Hanscom AFB, MA, March 1977, pp. 439.

Vick Charles R., "Specifications for reliable software," Proceedings of EASCON, Washington, DC, 7-9 October 1974.

Wasserman A.I., "Some Principles of User Software Engineering for Information Systems," IEEE COMPCON Spring '75, pp. 49-52.

Wasserman A.I., "A Top-Down View of Software Engineering," IEEE Proceedings of the 1st National Conference on Software

Engineering, Washington D.C., 11-12 September 1975, pp. 1-7.

Weiss D.M., The MUDD Report: A Case Study of Navy Software Development Practices, NRL Report 7909, Naval Research Laboratory, 21 May 1975, pp. 28.

Werson S.J., Cooper D.W., Stone R.L., "Hierarchical verification and validation software design," Vol. 2, CR-7-626, General Res. Corp., March 1976.

Whitaker W.A., "A Defense View of Software Engineering," IEEE Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, 13-15 October 1976, pp. 358-362.

Williams R., "Managing the development of reliable software," presented at the 1975 Int. Conf. Reliable Software, TRW Systems, Redondo Beach, CA 90278, April 1975.

Willis R.R., "DAS: An Automated System to Support Design Analysis," Hughes Aircraft Company, June 1978, pp. 17.

Willis R.R., "DAS- An Automated System to Support Design Analysis," Proceedings, Third International Conference on Software Engineering, Hughes Aircraft Company, Atlanta, GA, 10-12 May 1978, pp. 109-115.

Winchester J.W., "A System Engineering Language User Interface," Hughes Aircraft Company, Detroit, MI, 23 February 1978, pp. 7.

Winchester J.W., "URL Input Preprocessor," 26 July 1978, pp. 12.

Winchester J.W., "A Methodology for using CARA System as a System Engineering Language," Vol. 2, Book 2, Section 9, Final, Hughes Aircraft Company, 9 May 1978, pp. 17.

Wolverton R.W., "The Cost of Developing Large Scale Software," Practical Strategies for Developing Large Software Systems, Addison-Wesley Publishing Co., Horowitz Ellis, ed., 1975, pp. 73-100.

"XDD Handbook," Version 1, Report No. 6652/119F/01/C312, Corporate Data Administration, Xerox Square -052, 1 May 1978, pp. 100.

Yourdon E., Constantine L., Structured Design, Yourdon Inc., New York, NY, 1975.

Zempolich B.A., "Effective Software Management Requires Consideration of Many Factors," Defense Management Journal, Vol. II, No. 4, October 1975, pp. 8-12.

APPENDIX D

AUTOMATED REQUIREMENTS ENGINEERING TOOL CAPABILITIES LIST

The automated tool language (1.0) and analyzer (2.0) features described in this appendix provide a concise list of the automated requirements engineering tool capabilities which support the Requirements Engineering Guidebook (Volume III). These capabilities are also discussed in 4.2.2 of this report.

1. Language

   1.1 Objects (50 character names)

      1.1.1 Functional Requirements (functions)

      1.1.2 Constraint Requirements

         1.1.2.1 Performance
         1.1.2.2 Physical
         1.1.2.3 Operability
         1.1.2.4 Design
         1.1.2.5 Test (see 1.1.7)

      1.1.3 Sources

      1.1.4 Tracekeys

      1.1.5 Unique Identifiers (Analyzer generated and maintained, see 2.11)

      1.1.6 System External and Internal Inputs and Outputs (I/O)

         1.1.6.1 Physical Structure
         1.1.6.2 Logical Structure

1.1.7 Test Points

1.2 Relationships

1.2.1 Hierarchical Relation of Functional Requirements

1.2.2 Association of Constraint Requirements to Functional Requirements

1.2.3 Related-Requirement Relationship

1.2.4 Association of Sources to Requirements

1.2.4.1 Primary
1.2.4.2 Secondary

1.2.5 Association of Tracekeys to Requirements

1.2.6 Information-Flow Relationships

1.2.6.1 "Uses/used by" relationship
1.2.6.2 "Derives/derived by" relationship
1.2.6.3 "Updates/updated by" relationship
1.2.6.4 "Provides/Provided by" relationship
1.2.6.5 "Receives/Received by" relationship

1.2.7 Control-Flow Relationships

1.2.7.1 "Triggers/triggered by" relationship
1.2.7.2 "Conditional trigger/conditional triggered by" relationship (boolean AND, OR)
1.2.7.3 "Utilizes/utilized by" relationship

1.2.8 Hierarchical Relation of System Inputs/Outputs

1.3 Requirements Text

2. Analyzer (to be used interactively)

2.1 Change Requirements Data Base Reports

2.1.1 Input Report (update or no update feature)

2.1.1.1 Check syntax of requirements definition (consistency with language)

2.1.1.2 List (print) requirements definition and errors

2.1.1.3 Source input check option

Checks requirements entered into the requirements data base. Identifies illegal source references by comparison with a master list. Optionally notes source references which have been previously associated with other requirements. Names and lists source references for requirements with common references.

2.1.2 Delete Objects or Relationships Report

2.1.2.1 Check legality of deletion

2.1.2.2 Record deletions and errors

2.1.3 Rename Objects Report

2.1.3.1 Check legality of rename

2.1.3.2 Record renames and errors

2.1.4 Change Object Type Report

2.1.4.1 Check legality of change type
2.1.4.2 Record change types and errors

2.2 Object Information Report

Takes as input a list of objects. Prints requirements data base information concerning each object. The operator specifies the type of information desired, all requirements data base information relevant to each object is printed (default). The output is presented in the format of the input and delete relationship reports (2.1.1 and 2.1.2).

2.3 Source Document Summary Report

Prints requirements and relationships in the order which they appear in the source documents. Requirements having either a primary and or secondary relation to a source document are printed next to the source reference (with relation type specified) on a sequential list of all possible source references.

2.4 Functional Hierarchical Structure Report

Prints total or any part of functional hierarchical structure

2.4.1 Functional Hierarchical Structure

2.4.2 Information Available at Operator Option:

2.4.2.1 Sources (primary and secondary)
2.4.2.2 Associated constraints
2.4.2.3 I/O and relationships "uses", "derives", "updates"

2.4.2.4 Functional control relationships:
"triggers", "conditional trigger",
"utilizes"

2.4.2.5 Using activities and relationships:
"provides", "receives"

2.4.2.6 Requirements Text

2.5 I/O Hierarchical Structure Report

Prints any of the I/O structures below:

2.5.1 Prints an I/O structure beginning with a superior member of the structure

2.5.2 Prints an I/O structure beginning with a selected member of the structure

2.6 I/O-Function Interaction Report

Shows how a given collection of system input or output is "used", "derived", or "updated" by functions. Shows how a given collection of functions "uses", "derives", or "updates" system I/O. The report format is such that the analyst can trace information flows. The analyzer utilizes the defined I/O structure and breaks down high level I/O when needed to show information flows.

2.7 Information Flow Report

This report begins with a specified piece of system I/O. Shows the sequence of functions and intermediate I/O which (1) are necessary to derive an output from a system input, or (2) use this output to trace the information flow back to associated inputs. This report, like the I/O Function Interactions Report (2.6) utilizes the defined I/O interactions to show the continuous information flow. All test points on the flow paths are indicated.

2.8 Control Flow Report

Shows the flow of control following or leading to a given function. Differentiates between "triggers" and "conditional triggers". All conditions upon which the alternate paths are selected are associated with the conditional trigger. Shows any "utilizes" relationships. All test points on the flow paths are indicated.

2.9 Identify Specified Objects Report

Finds and lists the requirements data base objects which possess any combination of the characteristics specified below. Creates a file of object names which may be input for further analysis or report generation.

2.9.1 Identify Functional Requirements

2.9.1.1 All functions

2.9.1.2 Functional requirements below specified function in the hierarchy

2.9.1.3 Functional requirements with, or with no, associated constraint requirements

2.9.1.4 Functional requirements which "use" and/or "derive" and/or "update" a particular level in the I/O hierarchical structure

2.9.1.5 Functional requirements which do not "use" and/or "derive" and/or "update" a particular level in the I/O hierarchical structure

2.9.1.6 Functional requirements with any combination of associated control flow relationships: "triggers", "conditional triggers", "utilizes".

2.9.1.7 Functional requirements without any combination of associated control flow relationships: "trigger", "conditional triggers", "utilizes"

2.9.1.8 Functions at the lowest level in the hierarchy

2.9.1.9 Functions at the lowest level in the hierarchy below a specified function

2.9.1.10 Functional requirements without source references

2.9.2 Identify Constraints

2.9.2.1 All constraint requirements

2.9.2.2 Constraint requirements not associated with functions

2.9.2.3 All constraint requirements of a given type

2.9.3 Identify System I/O

2.9.3.1 All I/O, or any I/O below a specified level in the I/O hierarchy

2.9.3.2 I/O neither "used", "derived", nor "updated"

2.9.3.3 I/O contained between a particular section of the I/O structure

2.9.3.4 I/O "used" and/or "derived" and/or "updated" by a selected (as in 2.9.1.4 above) group of functions

2.9.4 Identify Source References

2.9.4.1 All source references

2.9.4.2 Source references without associated functional or constraint requirements

2.9.4.3 Source references with a specified prefix

2.9.4.4 Source references with a specified suffix

The prefix and suffix allow unique identifiers to be used for each unique source document referenced.

2.9.5 Identify Tracekeys

2.9.5.1 All tracekeys

2.9.5.2 Tracekeys without associated functional or constraint requirements.

2.9.6 Identify Test Points

2.10 Find Associated Requirements Report

This report begins with a list of requirements. For each of these requirements the analyzer finds requirements. Associated requirements satisfy any specified combination of the criteria described below. The nature of the relationship and source references are identified with the associated requirements.

2.10.1 Finds other requirements which "use", "derive", or "update" I/O relative to a specified input or output

2.10.2 Finds functions which "trigger", "are triggered by", or are "utilized by" a specified function

2.10.3 Finds requirements which are related by a constraint to function

2.10.4 Finds requirements which have source references (primary or secondary) common to a specified requirement

2.10.5 Finds requirements which are related by a Related-Requirement relationship

2.10.6 Finds requirements which are immediately superior in the functional hierarchy

2.11 Unique Requirement Number Derivation

Derives a unique requirement identifier. The function requirement number is related to the requirement position in the associated hierarchies. The numbers for constraint requirements are related to the functional hierarchical position of the related function(s).

2.12 Requirement Document Reports

Prints a list of the requirements together with the requirements text and other desired requirements data base information. The order of these requirements will follow that of the above derived numbers. Requirements data base information available optionally are source references and tracekeys.

2.13 Requirements Traceability Reports

Prints the types of traceability reports described below. The latter three reports use the tracekey to link the requirements in two different requirements data bases. The first report is the same as the Source Document Summary Report with the addition of unique identifiers (these two reports may be consolidated).

2.13.1 Shows traceability of source requirements (represented by a source reference) to requirements data base information (represented by a unique identifier and name)

2.13.2 Shows traceability of requirements data base information (represented by a unique identifier and requirement name) to requirements in a second requirements data base (represented by a requirement name and a source reference or unique identifier).

2.13.3 Shows traceability of source requirements (represented by a source reference and optionally the associated requirement name) to requirements in a second requirements data base (represented by a requirement name and a source reference or unique identifier).

2.13.4 Provides a list of requirements which do not trace to a second requirements data base. Associated with the requirement are the source references and the unique identifier. The list is arranged in the order of source references or in the order of unique identifiers as specified by the operator.

2.14 Test Reports

The test reports are used to evaluate the quality and completeness of test plans and procedures. Reports can be prepared for each test case defined for the information and control flows. The test reports show the relationship between test points and associated test cases, test plans, and procedures and other pertinent source documentation through the use of tracekeys.

2.15 Requirements Data Base Status Reports

The requirements data base status reports provide summary information on the contents of the requirements data base. Requirements data base objects are listed along with various statistics showing the quantities, percentages (as appropriate), and quality of each object in the requirements data base. Statistics presented for each object might include:

- Status of the requirement, such as complete, incomplete, inconsistent, ambigious, redundant (all entered into the requirements data base by analysts during the requirements engineering activities)

- Relationships between requirements such as sources/tracekeys and flow relationships (triggers, uses/derives/updates, etc.)

- Number of lines of descriptive text in the requirements data base

APPENDIX E

REQUIREMENTS ENGINEERING EXAMPLE

1. INTRODUCTION

The example contained in this appendix has been derived from the requirements engineering activities associated with an Air Force surveillance system acquisition. An excerpt from the surveillance system Type A segment specification has been included at the conclusion of this appendix. The requirements engineering activities described in this example are based on the statement of requirements contained in the excerpt, i.e. the excerpt is a source document. The example presents a description of the actual requirements engineering activities performed on the specification in conjunction with the use of an automated requirements engineering tool, Logicon-Extended CADSAT[1]. An automated requirements engineering tool like CADSAT helps the system engineer define a set of discrete system requirements. Discrete requirements are desired because they are non-redundant, testable, traceable, and communicable and are amenable to tests for completeness and consistency. Once the requirements have been tentatively defined, the tool helps the system engineer analyze for completeness and consistency. The tool can generate specification documents.

[1] The Computer-Aided Design and Specification Analysis Tool (CADSAT) is an Air Force owned requirements analysis tool developed by the University of Michigan under ESD/TOI contract F19628-76-C-0197. The extended version is a modification developed by Logicon for applications to military systems under ESD/OCU contract F19628-76-C-0218. CADSAT's User Requirements Language/User Requirements Analyzer (URL/URA) is basically equivalent to the Problem Statement Language and Problem Statement Analyzer (PSL/PSA) developed at the University of Michigan.

2. AUTOMATED TOOL CAPABILITIES

Automated tools like CADSAT assist requirements engineering in four ways:

- Provide a medium for formal requirements definition
- Perform rudimentary analysis
- Produce documentation
- Permit a flexible, iterative approach to requirements definition

Using a requirements engineering tool like CADSAT, the system engineer incorporates in a requirements data base certain salient features of each system requirement. The requirement is given a descriptive name. References to descriptions of the requirement are associated with each requirement (primary source). References to supplementary or related material may also be associated with the requirement (secondary source). The requirement is categorized as a functional requirement or one of several constraint requirements. A functional hierarchy is developed. Constraint requirements are associated with functional requirements. The flow of control (processing sequence) is defined. System information flows (I/O used and derived by system functions) are described. Finally, a comprehensive description of the requirement is incorporated in the requirements data base.

The automated tool allows the various system requirements to be incorporated in the requirements data base when they become known to the engineer. This is a prime advantage of an automated tool - the file of information about a system requirement is built up in an evolutionary manner. The requirements data base development proceeds in the same way that the system engineer naturally develops an understanding of the system. For example, when an additional reference to a previously identified requirement is found, this new reference is added to the requirements data base. If, during the course of an information-flow analysis, the engineer finds that he previously misunderstood the requirement or the specification developer receives feedback from the intended system users, appropriate corrections can be made to the requirements data base.

As the requirements analysis proceeds, the automated tool generates intermediate reports which help the systems engineer to complete and refine the requirements definition. These reports present, in various formats, the information which has been incorporated in the requirements data base. The reports present the results of various kinds of analysis. For example, one report lists selected requirements with associated source references or other requirements data base entries in a top-down hierarchical manner. Another report shows how the source documentation has been decomposed into requirements and points to those areas which have not been considered. Other reports show the control and information flows.

An automated tool like CADSAT can generate reports which provide high requirement visibility and provide final system documentation.

3. EXAMPLE REQUIREMENTS ENGINEERING PROCEDURE

Although a tool user could define all aspects of a requirement when that requirement is first identified, experience has shown that it is convenient to proceed in two stages. During the first stage the basics are defined. The requirements are identified, characterized and the functional hierarchy described. The control and system information flows are described during a second stage. This separation into phases is not hard and fast, and appropriate changes and additions are made at any time. The two-stage approach was applied in the example described in this appendix.

In the remainder of this appendix, cross references between the example description and the requirements engineering activities in the Requirements Engineering Guidebook (Volume III) are identified by references to appropriate activities (Volume III: BLOCKS 1-14).

3.1 Stage One

The first stage of the requirements data base development in this example consists of identifying the basic system functions, primary and secondary source references, organizing the functions into a functional hierarchy,

and identifying constraint requirements.

3.1.1 Identification of Functional Requirements (BLOCK 3)

One of the first steps in the development of a requirements data base is the identification of functional requirements (BLOCK 3). Each functional requirement is given a descriptive name. All appropriate source references are associated with the requirement (BLOCK 13). A complete statement of a functional requirement includes a description of the required function, a statement of when and under what conditions the function is activated, and a description of the system I/O that is to be transferred and the results. Source material frequently describes part of a requirement in one place and part in another. Sometimes minor variations in the activities of the function are discussed separately from the fundamental function. The conditions under which the activities of the function are to be performed may be discussed in the source material separate from the discussion of the function. All of this information about the requirement is consolidated in the requirements data base. References to source material which are necessary to completely describe the requirement are called primary references.

3.1.2 Identification of Secondary Source References (BLOCK 13)

Secondary source references are described along with the primary references.[1] The secondary references refer to information about closely related requirements, discussions of the rationale for the function or any other type of useful background information. This is information which will be reviewed when the requirement is incorporated into a specification (BLOCK 12) and might also be employed later to aid change impact analysis.

[1] The concept of secondary source references and other concepts presented in this appendix are a combination of current and proposed CADSAT capabilities resulting from the Requirements Standards Study (RSS).

3.1.3 Development of a Functional Hierarchical Structure (BLOCK 4)

The functional requirements are organized into a hierarchical structure (BLOCK 4). This structure shows how the functions break down from the general to the specific. Source references to descriptions of general functional capabilities are associated with the appropriate level in the hierarchy (BLOCK 13). Dummy headings (with no source reference) may be created where necessary to group a set of related requirements logically. The functional breakdown continues until a complete understanding of the preceding higher level function is achieved.

3.1.4 Identification of Constraint Requirements (BLOCK 5)

Constraint type requirements and their association with the appropriate functional requirements (BLOCK 5) are identified. The constraint requirements include performance, physical, operability, test and design requirements. Performance requirements are timing and accuracy considerations. Operability requirements include reliability, maintainability and availability considerations. An example of a physical constraint is a size limitation.[1] Source references are associated with constraints in the same manner as with functions.

3.2 Stage Two

The second stage of the requirements data base development consists of the formal description of system information (BLOCK 10) and control flows (BLOCK 9). This stage of the requirements data base development is normally performed after stage one is complete for the whole system. It has been found easier to perform the stage two analysis after the functional and constraint requirements have been tentatively identified and source references associated with the requirements. Stage two activities consist of the identification of system external inputs and outputs, the definition of information flow and of control flow.

1 A complete description of constraint requirements is contained in the RSS technical report (Volume III).

3.2.1 Identification of System External Inputs and Outputs (BLOCK 7)

Much of the I/O that is to be manipulated by the system can be identified by examining the information which is input to and output from the system. Input and output messages, operator inputs, and displays should be examined to identify the I/O handled by the system. Other system I/O which is known to the analyst from the stage one analysis should be formally identified and entered into the requirements data base at this time. Each input and output is given a short descriptive name. For communicability, the names should closely resemble those used in the source documentation. Not all system I/O will be recognized at this point. Additional I/O will be included at a later stage in the analysis.

A structure is provided for the I/O (BLOCK 8). There are two types of I/O structures: physical and logical. An example of a physical data structure is a message. A message is a collection of I/O which has a definite physically recognizable structure. In the example which follows, physical collections of information are called "entities". A logical I/O structure is simply a convenient collection of information. For example, one might wish to collect all kinds of weather information under the heading weather data.

The various kinds of weather information need not have any physical relationship. In this example organizational collections of I/O are called "sets".

3.2.2 Definition of Information Flow (BLOCK 10)

The system engineer considers each function to define the inputs received or used, and the outputs derived by that function. To aid this process the engineer employs the I/O hierarchies generated (BLOCK 8) and the functional hierarchy (BLOCK 4) constructed during the first stage of the requirements data base development as previously described. Since the I/O identification is probably not complete, some new I/O may be identified during information flow analysis. If the I/O used and derived by the

function is not clearly described in the source documentation then the requirements are not completely defined.

3.2.3 Definition of Control Flow (BLOCK 9)

The control flow shows the sequence of system processing. This sequence is specified in two ways. A "conditional trigger" is used to indicate a function that sometimes follows. The "trigger" specifies processing that necessarily follows. If the control flows cannot be defined from available source documentation then the requirements are not properly defined.

The definition of control flows may be facilitated by making use of the "utilizes" relationship. A primary function is said to "utilize" a secondary function when the secondary function supplies a result which is necessary for the operation of the primary function. The "utilizes" relationship is employed to simplify the illustration of control flows when the same function is required many times under different circumstances.

4. SAMPLE APPLICATION

The following is a detailed discussion of the requirements engineering activities performed for an excerpt of the surveillance system segment specification. It illustrates the requirements engineering activities of the two stages described above and includes selected CADSAT-like reports.[1] This example is cross referenced to the requirements engineering activities (BLOCKS 1-14) of the Requirements Engineering Guidebook (VOLUME III).

1 Some liberties in the format of the reports presented in this example have been taken to include recommended changes to CADSAT as discussed in the RSS technical report (Volume II).

4.1 Example of Stage One Procedure

An analyst convention used in this example to identify unnumbered paragraphs is as follows. Any paragraph which is followed by a series of unnumbered paragraphs (text separated from the preceding text with at least one blank line) is given an additional alphabetic character. This character is noted by the analyst in the left hand edge of each page in the source documents as shown in the excerpt (included at the end of the appendix). When developing a requirements data base, the source reference paragraph numbers include this additional character as a suffix to the actual paragraph number. For example, the third unnumbered paragraph after 3.7.1.2.7.2 becomes 3.7.1.2.7.2.c. Additionally, a one character prefix is included as a key which identifies the actual source document. In the following figures the paragraph numbers (set off by dashes instead of dots) would appear as r-3-7-1-2-7-2-c, etc.

As an example of the stage one procedure, consider how the requirements listed in paragraph 3.7.1.2.7.2 of the sample source material might be analyzed. The first step is to read the material and attempt to identify the required functions (BLOCKS 1 & 3).

Subparagraph "a" is a general description of the commitment function. Subparagraph "b" describes check tracking capacity and check guidance capacity functions and generate information for display function. Subparagraph "c" describes a status bypass option. Subparagraph "d" elaborates on the weapons team assignment process described in the previous paragraph. The last sentence describes a requirement to restrict the modification of the system data base. Subparagraph "e" describes the various recommit actions. The check guidance capacity function is described again. A restrict control function is also described.

Figure 1 shows a functional hierarchy (BLOCK 4) developed from the requirements of the sample paragraph, 3.7.1.2.7.2. Note that the hierarchy includes some requirements which are not explicitly mentioned in 3.7.1.2.7.2, and that some of the requirements, namely those discussing

trigger", "conditional triggers", "utilizes

line	level	name	reference
1	2	commit-recommit-to-mission	
2	3	commit-interceptors	
3	4	commit-to-cap	r-3-7-1-2-4-2-2-c
4	4	commit-to-intercept	r-3-7-1-2-4-2-2-c
5	3	chk-capacity-for-control	
6	4	chk-tracking-capacity	r-3-7-1-2-7-2-b r-3-7-1-2-4-2-3-b r-3-7-1-2-4-2-1-a
7	4	chk-channel-availability	r-3-7-1-2-7-2-e r-3-7-1-2-7-2-b r-3-7-12-7-3-2
8	3	recommit-interceptors	
9	4	recommit-to-intercept	r-3-7-1-2-7-2-e
10	4	recommit-to-cap	r-3-7-1-2-7-2-e
11	4	recommit-to-rtb	r-3-7-1-2-7-e

Figure 1. Surveillance System: (Partial) Functional Hierarchical Structure Report

each unique source document referenced.

system data base access, are not included here. The object of the hierarchy is to group logically the required functions for analysis purposes. Since the commit function is similar to the recommit function, it was decided to group these together. The two capacity checking functions are also included under the same heading because these functions are intimately connected with the commit-recommit functions. On the other hand, the restrict control function of paragraph "d" is included elsewhere in the hierarchy, since it involves the interceptor assignment function and is only indirectly related to interceptor commitment. Note that there are probably many other "logical" hierarchical arrangements of these requirements. The arrangement of requirements in a hierarchy is not meant to imply a particular design. The functional hierarchy is intended only to logically group functions for the purpose of defining the functional requirements in a convenient communicable form.

Figure 2 shows a printout of the requirements abstracted from 3.7.1.2.7.2 listed in the order that they appear in the source material. This type of report is used to check the completeness of the coverage of the source documentation (BLOCK 14). The requirements listed relative to each reference are examined to assure complete coverage of the topics discussed in a particular subparagraph. Subparagraphs which have been overlooked will be absent from this list.

Figure 3 shows the commit-recommit-to-mission section of the requirements hierarchy after additional requirements definition has been performed (BLOCK 3). Notice that several requirements have more than one primary source reference (BLOCK 13). Examination of additional source material has revealed supporting information or a more detailed description of some aspect of the requirement.

Figure 3 shows several other changes. A reference to 3.7.1.2.7.2a has been associated with the general requirement, commit-interceptors (BLOCK 13). This association is appropriate since the subparagraph provides a general description of the commitment function. An incorrect source reference to recommit-to-rtb has been corrected (BLOCK 14). An additional requirement,

REQ. NO.	REQUIREMENT NAME
3-7-1-2-7-2-a	
3-7-1-2-7-2-b	chk-tracking-capacity
	chk-channel-availability
3-7-1-2-7-2-c	
3-7-1-2-7-2-d	automatic-assign-track
	limit-data-access
3-7-1-2-7-2-e	recommit-to-intercept
	recommit-to-cap
	chk-channel-availability
	restrict-actions-to-assigned

Figure 2. Surveillance System: (Partial) Source Document Summary Report

line	level name	reference
1	2 commit-recommit-to-mission	
2	3 commit-interceptors	r-3-7-1-2-7-2-a
3	4 commit-to-cap	r-3-7-1-2-4-2-2-c
4	4 commit-to-intercept	r-3-7-1-2-4-2-2-c
5	3 chk-capacity-for-control	
6	4 chk-tracking-capacity	r-3-7-1-2-7-2-b
		r-3-7-1-2-4-2-3-b
		r-3-7-1-2-4-2-1-a
7	4 chk-channel-availability	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-2-b
		r-3-7-1-2-7-3-2
8	3 recommit-interceptors	
9	4 recommit-to-intercept	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-3-1-b
10	4 recommit-to-cap	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-3-1-a
		r-3-7-1-2-7-3-2-a
		r-3-7-1-2-7-3-2-b
		r-3-7-1-2-7-3-2-c
11	4 recommit-to-rtb	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-3-3
12	3 commit-with-status-bypass	r-3-7-1-2-7-2-c

Figure 3. Surveillance System: (Partial) Functional Hierarchical Structure Report including changes and additional primary source references

commit-with-status-bypass, has been added to the hierarchy (BLOCKS 3-4).

Figure 4 shows the commit-recommit-to-mission section of the hierarchy after it has been further refined. This report also lists the secondary source references for each requirement (BLOCK 13). (These are listed under the requirement to which they refer after "sref".) For an illustration of the use of secondary references, consider those associated with recommit-to-intercept (line 9). The reference 3.7.1.2.4.2.2c refers to a discussion of the commit-to-intercept requirement. Since the recommit-to-intercept and commit-to-intercept requirements are similar, this reference provides a useful perspective. "Sref 40-2-18" refers to a discussion of the recommit actions in an appendix to the segment specification. The last two references, 3.7.1.2.7.3.1d and "e" (not included in the specification excerpt) provide a redundant discussion of the processing which follows the recommit-to-intercept function. The information provided in these secondary references provide information which may be useful when the text of the requirement is rewritten for the specification under development. This information may also be utilized for the analysis of specification changes or engineering change proposals (BLOCK 13).

Two other changes are shown in the report in Figure 4. The name of the requirement chk-channel-availability has been changed to chk-guidance-capacity (Line 7). The change was made to better reflect the nature of the functional requirement and to indicate that the requirement is parallel to chk-tracking-capacity. The name of the requirement commit-with-status-bypass was changed to record-intcep-availability (line 11) to reflect a fuller understanding of the requirement. Examination of other parts of the specification revealed that the requirement stated in 3.7.1.2.7.2c is only part of a larger requirement, not explicitly stated, to maintain records of interceptor availability. (Those places in the source material which allude to the requirement are included as secondary references.) An explicit statement of this requirement should, of course, be included in the specification (BLOCK 12). Requirements like this, which are not fully understood initially, can be marked in various ways for later analysis. (Put -XX at the end of the name or provide a reference XX, for example.)

line	level names	reference
1	2 commit-recommit-to-mission	
2	3 commit-interceptors	r-3-7-1-2-7-2-a
3	4 commit-to-cap	r-3-7-1-2-4-2-2-c
	sref r-40-2-18	
	sref r-3-7-1-2-7-3-2-a	
	sref r-3-7-1-2-7-3-2-b	
	sref r-3-7-1-2-7-3-2-c	
4	4 commit-to-intercept	r-3-7-1-2-4-2-2-c
	sref r-3-7-1-2-7-2-b	
	sref r-3-7-1-2-4-2-2-a	
	sref r-3-2-1-3-1	
	sref r-40-2-18	
	sref r-3-7-1-2-4-2-2-j	
5	3 chk-capacity-for-control	
6	4 chk-tracking-capacity	r-3-7-1-2-7-2-b
		r-3-7-1-2-4-2-3-b
		r-3-7-1-2-4-2-1-a
	sref r-60-5-4	
7	4 chk-guidance-capacity	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-2-b
		r-3-7-1-2-7-3-2
	sref r-3-7-1-2-7-3	
	sref r-60-5-4	
8	3 recommit-interceptors	
	dsn base-stopr-capacity	r-3-7-1-2-7-2-e
9	4 recommit-to-intercept	r-3-7-1-2-7-2-e
		r-3-7-1-2-7-3-1-b
	sref r-3-7-1-2-4-2-2-c	
	sref r-40-2-18	
	sref r-3-7-1-2-7-3-1-d	
	sref r-3-7-1-2-7-3-1-e	
10	4 recommit-to-cap-rtb	r-3-7-1-2-7-2-e
	sref r-3-7-1-2-4-2-2-c	
	sref r-40-2-18	
	sref r-3-7-1-2-7-3-2-a	
	sref r-3-7-1-2-7-3-2-b	
	sref r-3-7-1-2-7-3-2-c	
	sref r-3-7-1-2-7-3-3	
11	3 record-intcep-availability	r-3-7-1-2-7-2-c
		r-3-7-1-2-7-8
	sref r-60-5-15	
	sref r-60-5-16	

Figure 4. Surveillance System: (Partial) Functional Hierarchical Structure Report including a design (dsn) constraint and secondary source references (sref).

The report in Figure 4 also shows the treatment of constraint requirements (BLOCK 5). A requirement for the program to accomodate 15 bases and 60 Strategic Orbit Recovery Points (STOPRs) is specified in subparagraph "e" of 3.7.1.2.7.2 in source material. This is a design requirement which refers to the interceptor recommitment function. This constraint requirement is called base-stopr-capacity and has been associated with the function recommit -interceptors (line 8). Design requirements in this example report have been given the abbreviation "dsn".

4.2 Example of Stage Two Procedure

The report in Figure 4 provides the system engineer with source documentation references to information needed to define information flows (BLOCK 10) and control flows (BLOCK 9).

4.2.1 Identification of System External Inputs and Outputs (BLOCK 7)

The first step of this stage of the analysis is to identify the major features of the system I/O and to rough out I/O structures. The inputs to the commit-interceptors functions (lines 3 and 4 of Figure 4) are described in paragraphs 3.7.1.2.4.2.2 "a" and "c". Among these inputs are a squadron identifier, interceptor callsign, departure airbase, the mission and a stopr. Mission need not be specified as a system I/O at this time, since the mission is specified by the operator's selection of function and will be identified as part of control flow (BLOCK 9). These inputs are gathered together as commit-recommit-inputs in the I/O structure (Figure 5). Since there is no physical relationship among these inputs, they are grouped into a logical collection of data (a set). In addition to the above inputs, some means to identify a target must be provided. The target is identified with a huk track number. (huk is hostile (h), unknown (u), faker (k).) The huk track number is meaningful only in association with other huk track data. Therefore, in the I/O hierarchy, huk-track-number is included as part of the I/O structure, huk-track-data (an entity, 4*).

The outputs of the commit functions are not obvious from the source material. The purpose of the commit functions is to associate information

```
1*    1 commit-recommit-inputs (set)
 1      2 departure-base
 2      2 stopr
 3      2 interceptor-callsign
 4      2 squadron-designator
 5      2 recovery-base
2*    1 manned-interceptor-data (entity)
 1      2 interceptor-track-number
 2      2 departure-base-number
 3      2 interceptor-mission-data
 4        3 mission-indicator
 5        3 cap-rtb-data
 6          4 stopr-number
 7          4 recovery-base-number
 8        3 target-data
 9          4 target-track-number
3*    1 interceptor-track-data (entity)
 1      2 interceptor-track-number
4*    1 huk-track-data (entity)
 1      2 huk-track-number
```

Figure 5. Surveillance System: (Partial) I/O Hierarchical Structure Report.

relevant to a particular interceptor when that interceptor is commited. This information includes the interceptor track number, information about the mission and the target, guidance data, fuel data, etc. The entity manned-interceptor-data (Figure 5, 2*) is this collection of interceptor data. Specifically, some of the system data produced by the commit functions are the interceptor-track-number, departure-base-number, a mission-indicator and stopr-number. In general it is best to describe system I/O in the terms used by the documentation. In this case, however, it was necessary to create -indicators and -numbers in order to illustrate the nature of the functions.

Consider the recommit-interceptors functions (lines 9 and 10 of Figure 4). Paragraph 3.7.1.2.7.2c describes the inputs to these functions. All but the recovery base have already been described as inputs to the commit functions. The capacity checking and record keeping functions of Figure 4 are ancillary to the major commit-recommit functions and need not be considered for additional I/O at this point. The I/O structure of Figure 5 is sufficient as a basis for further analysis.

4.2.2 Definition of Information Flow (BLOCK 10)

The next step in the analysis is the detailed definition of the flow of I/O through each of the functions. Much of the needed I/O has already been described (Figure 5).

Consider the commit-to-cap function of Figure 4 (line 3). This function uses interceptor-callsign to derive the interceptor-track-number (3.7.1.2.4. 2.2a). It uses the stopr to derive the stopr-number (3.7.1.2.4.2.2c), and the departure-base to derive the departure-base-indicator (3.7.1.2.4.2.2c). The squadron-designator may also be used to derive the interceptor-track-number (3.7.1.2.4.2.2c). Additional possible inputs to the commit-to-cap function are described in another source paragraph, 3.7.1.2.7.3.2 (not included in the specification excerpt). The function may use a manual-heading-input to derive a manual-heading-indicator and a manual-heading. These data were not recognized at the time

of the development of I/O structure in Figure 5 and would be included in a later update. The function also derives the mission-indicator.

The commit-to-intercept function (Figure 4, line 4) uses and derives much of the same I/O as the commit-to-cap function (Figure 4, line 3). In place of the stopr, this function takes as input a huk-track-number (Figure 5). Instead of a stopr-number the function derives a target-track number (Figure 5).

The recommit functions use and derive much less I/O than the commit functions since interceptors which are recommited are already followed by the system. The recommit-to-intercept function (Figure 4, line 14) uses the huk-track-number to derive a target-track-number. The function also derives a mission-indicator. The recommit-to-cap-rtb function (Figure 4, line 10) uses a stopr to derive a stopr-number. It also uses the recovery-base or departure-base to derive a recovery-base-number. The mission-indicator is also derived.

The record-intcep-availability (record the interceptor availability) function (Figure 4, line 11) keeps records of the status of the interceptors at the various bases. To maintain this record, the function uses the departure-base-number and the interceptor-track-number. The function derives information which the analyst calls interceptor-availability-data. Because this availability data is not described in detail in the source documentation, it is not broken down in detail in the requirements data base. The status of the interceptors may be changed by a tactical-action update (3.7.1.2.7.8).

The source documentation discusses no I/O in connection with the chk-tracking-capacity and chk-guidance-capacity functions (lines 6 and 7 of Figure 4). The I/O associated with these functions is common only to the capacity checking and commit-recommit functions. This I/O does not appear elsewhere in the system. Since no purpose is served, the information flow is not defined for the capacity checking functions. The close association between the commit-recommit functions and the capacity checking functions is

adequately described using the "utilizes" relationship which is discussed below.

The information flows are summarized in the Hierarchical Structure report in Figure 6.

The matrix in Figure 7 shows the information flow at a glance. Both this matrix and the type of Functional Hierarchical Structure report shown in Figure 6 are reviewed during the information flow analysis (BLOCK 10) to assure completeness of I/O definition and consistency of flow (BLOCK 14).

4.2.3 Definition of Control Flow (BLOCK 9)

The purpose of the control flow analysis (BLOCK 9) is to assure that the sequence of of system operations is completely and consistently described. The procedure followed is similar to the definition of information flows.

Consider the commit functions of Figure 4 (lines 3 and 4). The initiate-interceptors function is triggered by commit actions (3.7.1.2.7.8). Several other functions (not discussed in the source material within this appendix) are triggered by the commit functions: estimate-intcep-spd-head-alt (estimate interceptor speed, heading and altitude), auto-select-profile and select-attack-option. The latter function, select-attack-option, is triggered only by the commit-to-intercept function since there is no attack during a cap (combat air patrol).

The commit functions utilize chk-tracking-capacity (3.7.1.2.7.2b), chk-guidance-capacity (3.7.1.2.7.2b) and automatic-assign-track (3.7.1.2.7.2d). These functions are said to be utilized by the commit functions since they represent functions which are required to complete the commit function activities. Functions which appear in many places in the flow sequence, such as the capacity checking functions, are often said to be utilized by other functions. This type of treatment simplifies the description of the

```
line    (level or relationship) names

 1      2 commit-recommit-to-mission
          dsn base-stopr-capacity
 2          3 commit-interceptors
 3            4 commit-to-cap
                uses manual-heading-input
                uses departure-base
                uses stopr
                uses interceptor-callsign
                uses squadron-designator
                trgs initiate-interceptors
                trgs record-intcep-availability
                trgs estimate-intcep-spd-head-alt
                trgs auto-select-profile
                utls chk-tracking-capacity
                utls chk-guidance-capacity
                utls automatic-assign-track
                drvs interceptor-track-number
                drvs mission-indicator
                drvs stopr-number
                drvs manual-heading-indicator
                drvs manual-heading
                drvs departure-base-number
 4            4 commit-to-intercept
                uses squadron-designator
                uses interceptor-callsign
                uses departure-base
                trgs initiate-interceptors
                trgs estimate-intcep-spd-head-alt
                trgs record-intcep-availability
                trgs select-attack-option
                trgs auto-select-profile
                uses huk-track-number
                utls chk-tracking-capacity
                utls chk-guidance-capacity
                utls automatic-assign-track
                drvs interceptor-track-number
                drvs mission-indicator
                drvs target-track-number
                drvs departure-base-number
```

Figure 6. Surveillance System: (Partial) Functional Hierarchical Structure Report including information and Control Flow relationships.

the rows are I/O, the columns are function names.

row names (I/O)

1 interceptor-track-number
2 mission-indicator
3 stopr-number
4 manual-heading-indicator
5 manual-heading
6 departure-base-number
7 manual-heading-input
8 squadron-designator
9 departure-base
10 stopr
11 interceptor-callsign
12 target-track-number
13 huk-track-number
14 interceptor-availability-data
15 tactical-action-update
16 recovery-base
17 recovery-base-number

column names (Function Names)

1 chk-capacity-for-control
2 chk-guidance-capacity
3 chk-tracking-capacity
4 commit-interceptors
5 commit-to-cap
6 commit-to-intercept
7 recommit-interceptors
8 recommit-to-intercept
9 recommit-to-cap-rtb
10 record-intcep-availability

Figure 7. Surveillance System: (Partial) Information - Function Interaction Report

(i,j) value	meaning
r	row i is received or used by column j (input)
u	row i is updated by column j
d	row i is derived or generated by column j (output)

```
                          COLUMN

                       (FUNCTIONS)

                                   1
                      1 2 3 4 5 6 7 8 9 0
                    +---------+-------+-------+
                  1 :         : d d   :     r :
                  2 :         : d d   : d d   :
                  3 :         : d     :   d   :
                  4 :         : d     :       :
                  5 :         : d     :       :
                    +---------------------------+
      ROW         6 :         : d d   :   r r :
                  7 :         : r     :       :
     (I/O)        8 :         :   r   :       :
                  9 :         : r r   :       :
                 10 :         : r     :   r   :
                    +---------------------------+
                 11 :         : r r   :       :
                 12 :         :   d   : d     :
                 13 :         :   r   : r     :
                 14 :         :       :     d :
                 15 :         :       :     r :
                    +---------------------------+
                 16 :         :       :   r   :
                 17 :         :       :   d   :
                    +---------------------------+
```

information-function interaction matrix

Figure 7 (cont'd)

control flow.

The recommit functions (lines 9 and 10 of Figure 4) trigger many of the same functions as the commit functions. The recommit functions, however, do not trigger initiate-interceptors since recommited interceptors are already tracked (and therefore have already been initiated). The recommit functions also do not trigger the record-intcep-availability function because a recommitment does not affect the status of an interceptor.

The recommit function utilizes the chk-guidance-capacity and restrict-intcep-data-access functions. The chk-tracking-capacity function is not utilized on recommitment, since recommited interceptors are already being tracked (3.7.1.2.7.2d). The restrict-intcep-data-access function is utilized here because the recommited interceptor is already followed by sytem I/O and it must be assured that the recommit is made by authorized personnel (3.7.1.2.7.2e).

The control flow description is summarized with the information flow description in Figure 6.

Figure 8 shows sample control flow report. The figure illustrates the sequence of processing which follows the commit-to-intercept function. This report shows how one of the functions discussed in this example is linked to functions of other parts of the system. Flow of control proceeds from left to right in the figure. Conditions which determine the flow direction (conditional triggers) are not represented in this figure.

```
                   +-----------+
                   :estimate-i-:
                   :ntcep-spd--:
                  *:head-alt   :
                   +-----------+
                 *
                   +--process--+
                *  :initiate-i-:
                   :nterceptor-:
              *  * :s          :
               *   +-triggered-+
              *
+--process--+      +--process--+      +--process--+      +--process--+
:commit-    : *    :auto-      :      :generate-  :      :make-      :
:to-        : *****:select-    : *****:guidance-  : *****:magnetic-  :
:intercept  : *    :profile    :      :solution   :      :corrections:
+-----------+      +-triggered-+      +-triggered-+      +-triggered-+
              *
               *   +--process--+      +--process--+      +--process--+      +--process--+
              *  * :select-    :      :select-    :      :select-    :      :generate-  :
                   :attack-    : *****:attack-    : *****:attack-    : *****:guidance-  :
               *   :option     :      :altitude   :      :speed      :      :solution   :
                   +-triggered-+      +-triggered-+      +-triggered-+      +-triggered-+
                *
                   +--process--+
                  *:record-int-:
                   :cep-availa-:
                   :bility     :
                   +-triggered-+
```

Figure 8. Surveillance System: (Partial) Control-Flow Report, Flow of control is from left to right.

SOURCE DOCUMENT EXERPTS

This excerpt contains selected source documentation used in the preceding example.

3.7.1.2.4.2.2 Interceptor track initiation. Initiation of interceptor tracks shall be accomplished by means of a SD/WAO/WD Commit action which includes two letters identifying a squadron or a complete interceptor callsign. (See Track Number in 20.2 for legal values.)

o

o

o

A commit action must include a departure airbase and mission (Integration against a specified target, CAP against a specified Strategic Orbit/ Recovery Point (STOPR), or CAP). The action may include an interceptor call-sign (Commit with track number) or the two letter squadron designator (Commit with squadron). Live interceptors shall utilize squadron call letters and values from 01 to 31 inclusive and simulated interceptors shall utilize the squadron call letters and values from 32 to 63 inclusive.

o

o

o

In response to an interceptor commitment the TG shall determine if there is any remaining track capacity. If the mission is interception or CAP to a STOPR, the TG shall also determine if there is any remaining interceptor guidance capacity. If capacity remains, the track shall be initiated and track status and position shall be set to Scrambled at the specified airbase; otherwise a Track of Interceptor Overload condition shall be generated for display to the operator taking the action.

AD-A077 848   LOGICON INC LEXINGTON MA   F/G 5/1

REQUIREMENTS STANDARDS STUDY. VOLUME II. TECHNICAL REPORT.(U)

OCT 79   D G SMITH, P B MERRITHEW   F30602-77-C-0207

UNCLASSIFIED   DSJ-R79007-VOL-2   RADC-TR-79-240-VOL-2   NL

3 OF 3

AD A077848

END

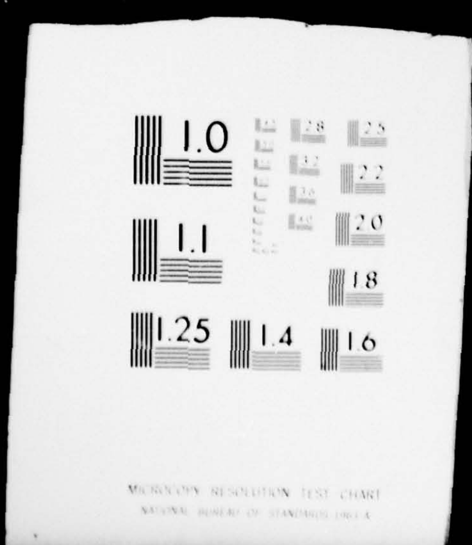
DATE FILMED

1-80

DDC

3 OF 3

AD
A077848



a 3.7.1.2.7.2 I
by which the
and thereby al
an interceptor

b The Tracking (
with 3.7.1.2.
shall be rej
interceptors
channel. Th
Scramble track

c The capabilit
using a Statu
departure squa

d The ICG sha
information a
in 3.7.1.2.7
interceptor,
for the targe
member taking
capable of mo

e The ICG shal
existing int
or with a sp
Air Patrol (
also process
RTB mission,
none has bee
rejected if
occupy a gui
The program
member Recom
the weapons

Interceptor commitment. Interceptor commitment is the process e weapons team member assigns an interceptor to a mission allocates computer capacity to track and generate guidance for or; the action results in a Scrambled interceptor.

Group will process weapons team Commit actions in accordance 2.4.2.2, establishing a new interceptor track. The action ejected if there is no unused tracking channel or, for requiring guidance (see 3.7.1.2.7.3), no unused guidance The ICG shall then generate the necessary information for ack and tabular displays.

ity shall be provided for the weapons team member to Commit tus Bypass option so that the interceptor status count for the quadron/airbase combination is not affected.

hall be provided for the weapons team member assignment as part of any weapons team member Commit action as specified .7.1. If no weapons team member is specified for a new , it shall be assigned to the weapons team member responsible get or, if none is assigned to the target, to the weapons team ng the action. Only the assigned weapons team member shall be modifying the data base associated with an assigned interceptor.

all process weapons team member Recommit actions, pairing an terceptor track with another track for an intercept mission, specified Strategic Orbit Recovery Point (STOPR) for a Combat (CAP) mission or Return-to-Base (RTB) mission. The ICG shall s Recommit to CAP or RTB actions with no STOPR specified; for an , the interceptor will be paired with the recovery base or, if een specified, with the departure base. The action shall be f the interceptor will require guidance, does not currently idance channel, and no unused guidance channel is available. shall accommodate 15 bases and 60 STOPRs. A weapons team mmit shall be processed only if the interceptor is assigned to team member taking the action.

3.7.1.2.7.3.2 Cap Mission. An interceptor on CAP is assigned to patrol over an area. It does not have a specific target assigned, but is considered to be a weapons source for future Recommitments. Interceptors on CAP do not necessarily occupy guidance channels. An Interceptor on CAP enters a guidance channel when it is Commited or Recommited to a STOPR, or when the assigned controller enters a manual heading. The insertion of a manual command heading shall be rejected if a guidance channel is not available. The insertion of any command data other than a manual command heading shall be illegal if the interceptor is not in a guidance channel. Processing for interceptors on CAP in a guidance channel shall be as follows:

CAP Manual Heading Inserted. Command heading shall be set to the values entered by the assigned controller. Guidance commands shall be generated as appropriate and full performance prediction shall be performed using the optimum approach speed and optimum approach altitude stored for the interceptor type unless overridden by operator action entry.

CAP Paired with STOPR. Guidance commands shall be generated as specified in 3.7.1.2.7.3.1a through d, and full performance prediction shall be performed. The interceptor shall be removed from the guidance channel and the pairing shall be broken when the interceptor is W15 nmi from the STOPR, except when a manual heading has been inserted.

3.7.1.2.7.3.3 RTB Mission. An interceptor is placed on RTB as a result of the Recommit to RTB action. Processing for interceptors on RTB shall be as specified for CAP -- Paired with STOPR missions, with guidance limited to the climb and cruise phases.

o

o

o

n and interceptor status. The ICG shall process and maintain a count of kills reported against well as for the Region. On Told-In tracks, kill information to the Information Transfer n in a Telling Control (Tactical Action Update) aintain counts of alert, airborne ordered and squadron-airbase pair and for the Region. In ion, separate tactical action and interceptor intained for real and simulated data. Counts dance with Telling Control (Tactical Action tus reports from the Information Transfer Group ion.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.